

04443351 **Image available**

METHOD AND APPARATUS FOR PROCESSING IMAGE BASED ON GRAPHICS LANGUAGE

PUB. NO.: 06 -087251 ([JP 6087251] A)
PUBLISHED: March 29, 1994 (19940329)
INVENTOR(s): RUISU TORABU PARUDO
HORUHE FUIRITSUPUSU
APPLICANT(s): CANON INC [000100] (A Japanese Company of Corporation),
(Japan)
APPL. NO.: 05-143425 [JP 93143425]
FILED: June 15, 1993 (19930615)
PRIORITY: 7-898,532 [US 898532-1992], US (United States of America),
June 15, 1992 (19920615)
7-911,030 [US 911030-1992], US (United States of America),
July 08, 1992 (19920708)
INTL CLASS: [5] B41J-029/38; B41J-002/52; G06F-003/12; G06F-015/66;
G06F-015/72
JAPIO CLASS: 29.4 (PRECISION INSTRUMENTS -- Business Machines); 45.3
(INFORMATION PROCESSING -- Input Output Units); 45.4
(INFORMATION PROCESSING -- Computer Applications)
JAPIO KEYWORD: R002 (LASERS); R105 (INFORMATION PROCESSING -- Ink Jet
Printers)
JOURNAL: Section: M, Section No. 1630, Vol. 18, No. 348, Pg. 75, June
30, 1994 (19940630)

ABSTRACT

PURPOSE: To enhance versatility through improvement in the performance of a printer without increasing physical resources by forecasting necessary resources in advance of print processing and changing the print processing flexibly on the basis of the forecast results.

CONSTITUTION: Printer resources required for the conversion into a full raster image can be accurately forecast by the formation of the code expression of an image and a dynamic adaptation of resources technique (DART). A control unit 1 inputs image data from a communication port 12, and a GL conversion unit 42 converts PDL image data into GL image data. A DART code formation unit forms DART expression which is stored in a memory 44. Resources required for the development of the full raster image are forecast here. For example, when CPU 41 has judged that printer resources were inadequate, the reassignment of resources is conducted, and the DART expression is modified to be accommodated within the range of the printer resources. When memory resources are in shortage, the expression is changed to another one of less memory.

Best Available Copy

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平6-87251

(43)公開日 平成6年(1994)3月29日

(51)Int.Cl.⁵

識別記号

庁内整理番号

FI

技術表示箇所

B 4 1 J 29/38

Z 9113-2C

2/52

G O 6 F 3/12

C

15/66

3 3 0 A 8420-5L

7339-2C

B 4 1 J 3/ 00

A

審査請求 未請求 請求項の数190(全 34 頁) 最終頁に続く

(21)出題番号

特願平5-143425

(22)出願日

平成5年(1993)6月15日

(31)優先權主張番号 07/898532

(32)優先日 1992年6月15日

(33)優先權主張国 米国 (U S)

(31)優先権主張番号 07/911030

(32)優先日 1992年7月8日

(33)優先権主張国 米国 (U S)

(71)出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72)発明者 ルイス トラブ バルド

アメリカ合衆国 カリフォルニア州

94306. バロ アルト, アマランタ

4090

(72)発明者 ホルヘ フィリップス

アメリカ合衆国 カリフォルニア州

94025, メンロ パーク, クラウド

アベニュー 970

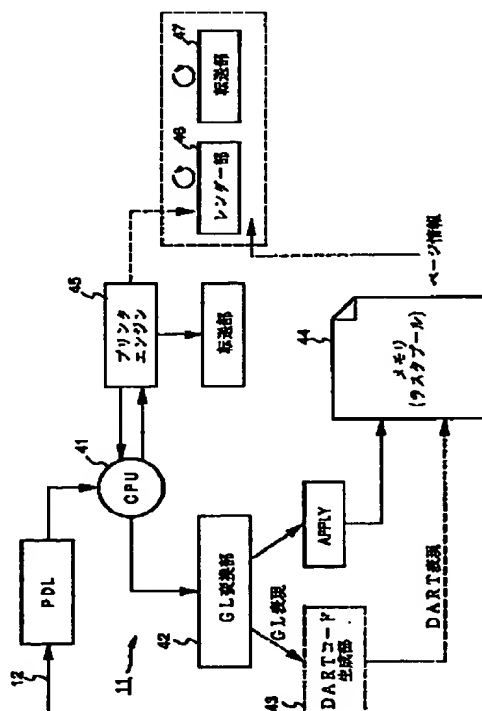
(74)代理人 弁理士 大塚 康徳 (外1名)

(54)【発明の名称】 グラフィックス言語に基づく画像処理方法及び装置

(57) 【要約】

【目的】 本発明の目的は、プリンタ内の資源をより効率的に利用することで、利用可能な物理的な資源を増加させることなくプリンタの性能を向上させることにある。

【構成】 装置に依存しないプリントコマンドを含むプリントコマンドに基づいて、グラフィックス言語文書記述コマンドを生成する方法と装置が提供される。プリント処理に必要な資源を予測し、予測された資源がプリンタ装置の資源の範囲内であるかどうかを判断し、プリント装置の資源が現時点でのプリント処理には足りないと判断された場合にプリント処理を変更し、現存資源の範囲内で画像のプリントを可能とするようなプリント装置が提供される。



【特許請求の範囲】

【請求項1】 プリント処理の開始に先立って必要となる資源を予測し、予測結果に基づいて適応的にプリント処理を変更することを特徴とするプリント装置。

【請求項2】 前記プリント処理はプリントデータの符号化表現をラスタ画像に展開する処理を含み、前記予測処理は符号化表現に基づいて行われることを特徴とする請求項1記載のプリント装置。

【請求項3】 前記プリント処理は予測結果に基づいて適応的に画質を変化させることを特徴とする請求項1記載のプリント装置。

【請求項4】 プリント処理において必要となる資源を予測する予測手段と、

該予測手段によって予測された資源がプリント装置の資源の範囲内であるか否かを判断する判断手段と、

前記判断手段によってプリント処理に必要な資源が足りないと判断された場合には、プリント処理を変更する変更手段とを備えることを特徴とするプリント装置。

【請求項5】 プリント処理中に画像の符号化表現を形成する手段をさらに備え、符号化表現をフルスタ画像に展開する間に前記予測処理が必要となる資源を予測することを特徴とする請求項4記載のプリント装置。

【請求項6】 プリント処理はプリント情報の各ページ中のバンドごとに行われ、前記予測手段は必要となる資源を各バンドごとに予測することを特徴とする請求項4記載のプリント装置。

【請求項7】 前記変更手段に基づくプリント処理の変更に応じて前記予測手段は必要となる資源を再予測し、前記変更手段は該予測手段によって予測された資源がプリント装置の範囲内に収まるまでプリント処理を繰り返し変更することを特徴とする請求項4記載のプリント装置。

【請求項8】 前記変更手段は前記プリントでプリントされる画像の質を変化させてプリント処理を変更することを特徴とする請求項4記載のプリント装置。

【請求項9】 解像度を変化させることを特徴とする請求項8記載のプリント装置。

【請求項10】 カラー解像度もしくは中間調レベル数を減少させることを特徴とする請求項8記載のプリント装置。

【請求項11】 一定の速度でプリント情報をスキャンしてプリントするプリントエンジンと、少なくとも一部のプリント情報を一定速度で出力することのできるビットマップメモリとをさらに備えることを特徴とする請求項4記載のプリント装置。

【請求項12】 前記予測手段は、画像を前記ビットマップメモリに生成するのに必要となる演算能力、画像を記憶するのに必要となるメモリ量、画像をビットマップメモリに転送するのに必要となるバンド幅のうちの少なくとも1つを予測することを特徴とする請求項11記載

のプリント装置。

【請求項13】 一定速度でプリント画像をスキャンして生成するプリントエンジンと、

少なくとも第1部位と第2部位とに適応的に分割されるメモリであって、第1部位はプリント画像の符号化表現を記憶し、第2部位は少なくとも一部分のラスタ画像を記憶するようなメモリと、

入力画像データをプリント画像の符号化表現に翻訳し、符号化表現をメモリの第1部位に記憶するような翻訳手段と、

該メモリの第1部位中に記憶されている符号化表現をフルスタ画像に展開するのに必要な資源を予測する予測手段と、

該メモリの第1部位中に記憶されている符号化表現をラスタ画像として該メモリの第2部位に展開する展開手段とを備えることを特徴とするプリント装置。

【請求項14】 前記予測手段によって符号化表現をフルスタ画像に展開するのに必要な資源が足りないと判断された場合には、プリント処理を変更する変更手段をさらに備えることを特徴とする請求項13記載のプリント装置。

【請求項15】 前記変更手段は、必要となるプリント装置が異なるように符号化表現を別の表現に変更することを特徴とする請求項14記載のプリント装置。

【請求項16】 前記変更手段はプリント解像度を変化させることを特徴とする請求項14記載のプリント装置。

【請求項17】 前記変更手段はカラー解像度もしくは中間調レベル数を減少させることを特徴とする請求項14記載のプリント装置。

【請求項18】 前記予測手段は、少なくとも符号化表現をラスタ画像に展開するのに必要な演算能力と、符号化表現を記憶するのに必要なメモリ量とを予測することを特徴とする請求項13記載のプリント装置。

【請求項19】 前記翻訳手段によって翻訳される画像データを入力する入力手段をさらに備えることを特徴とする請求項13記載のプリント装置。

【請求項20】 画像データはページ記述言語で入力されることを特徴とする請求項19記載のプリント装置。

【請求項21】 前記メモリの第1部位は符号化表現をバンドごとに記憶し、前記メモリの第2部位は対応するバンド情報のラスタ画像を記憶することを特徴とする請求項13記載のプリント装置。

【請求項22】 前記翻訳手段は入力画像データをバンドで翻訳することを特徴とする請求項21記載のプリント装置。

【請求項23】 前記予測手段はプリント情報の各バンドごとに必要となる資源を予測することを特徴とする請求項22記載のプリント装置。

【請求項24】 前記判断手段によって資源が足りない

と判断された場合には、前記展開手段はフルスタ画像を前記メモリの第1部位と第2部位のものと位置に展開することを特徴とする請求項23記載のプリント装置。

【請求項25】 前記翻訳手段は入力情報をバンド符号化ラスタ表現に翻訳し、前記判断手段が前記プリントエンジンの一定速度に対応する所定の時間間隔でバンド情報を展開するのに必要な演算能力が足りないと判断した場合に、該展開手段がバンドを前記メモリの第3部位に前もってラスタリングすることを特徴とする請求項13記載のプリント装置。

【請求項26】 前記予測手段によって符号化表現の方が対応するラスタ画像以上にメモリ量を必要とすると判断された場合には、前記変更手段がプリント処理を変更することを特徴とする請求項13記載のプリント装置。

【請求項27】 前記判断手段によって1つのバンド情報に必要な資源が足りないと判断された場合には、それ以外のバンドはそのまま、当該バンド情報のみを変化させることを特徴とする請求項13記載のプリント装置。

【請求項28】 前記表現は、ビットマップオブジェクト、ラン長オブジェクト、台形オブジェクト、ボックスオブジェクト、固定境界符号オブジェクトの少なくとも1つから成ることを特徴とする請求項13記載のプリント装置。

【請求項29】 前記展開手段は、アプリケーション機能によって、オブジェクトを背景に対して適用して、表現のラスタライズを行うことを特徴とする請求項28記載のプリント装置。

【請求項30】 第1部位と第2部位とに適応的に分割されるメモリであって、第1部位は画像データの符号化表現を記憶し、第2部位は少なくとも1つのバンドのラスタ画像データを記憶するメモリと、複数のバンドの各画像データを画像の符号化表現に翻訳し、該メモリの第1部位に符号化表現を記憶する手段と、

前記メモリの第1部位に記憶されている符号化表現に基づいて、バンド符号化表現を展開するのに必要な資源を予測する予測手段と、

前記予測手段によって必要となる資源がプリント資源では足りないと予測された場合に、該メモリの第1部位の符号化表現を変更する手段と、

前記メモリの第1部位に記憶されている符号化表現のバンドをラスタ画像データのバンドに展開し、該ラスタ画像データのバンドを該メモリの第2部位に記憶する展開手段と、

少なくとも該メモリの第2部位に記憶されているラスタ画像データをプリントするプリントエンジンであって、該プリントエンジンは一定速度でラスタ画像データのバンドをプリントするプリントエンジンとを備えることを

特徴とするプリント。

【請求項31】 入力画像データをページ記述言語で入力する入力手段をさらに備えることを特徴とする請求項30記載のプリント。

【請求項32】 前記翻訳手段は、ページ記述言語を、ビットマップオブジェクト、ラン長オブジェクト、台形オブジェクト、ボックスオブジェクト、固定境界符号オブジェクトの少なくとも1つを含む符号化表現に変換することを特徴とする請求項31記載のプリント。

10 【請求項33】 前記展開手段は、アプリケーション機能によってオブジェクトを背景上に対して適用し、画像をラスタリングする手段を備えることを特徴とする請求項32記載のプリント。

【請求項34】 前記予測手段は、符号化表現を構成するプリミティブに基づいて必要となる資源を予測することを特徴とする請求項32記載のプリント。

【請求項35】 前記予測手段は画像の各バンドごとに必要となる資源を予測することを特徴とする請求項30記載のプリント。

20 【請求項36】 前記予測手段は、演算能力、メモリ能力、バンド幅の少なくとも1つを予測することを特徴とする請求項30項記載のプリント。

【請求項37】 前記変更手段は、演算能力の資源が過大である場合に、より多くのメモリを用いるような符号化表現に変更することを特徴とする請求項36記載のプリント。

【請求項38】 前記変更手段は、前記展開手段でもってバンド画像データを前もってラスタリングすることを特徴とする請求項37記載のプリント。

30 【請求項39】 前記変更手段は画像の質を変化させることを特徴とする請求項37記載のプリント。

【請求項40】 前記変更手段は、必要となる資源がプリント資源以上となるバンドに対しては少なくとも前記展開手段によって画質を変化させることを特徴とする請求項39記載のプリント。

【請求項41】 前記変更手段は、前記展開手段によって全体の画像の質を変化させ、変化させた画像を前記符号化表現によってもとの位置に展開することを特徴とする請求項40記載のプリント。

40 【請求項42】 前記画質の変化は、画素密度、カラー解像度、グレイレベル数の少なくとも1つに対して行われることを特徴とする請求項40記載のプリント。

【請求項43】 前記画質の変化は、画素密度、カラー解像度、グレイレベル数の少なくとも1つに対して行われることを特徴とする請求項41記載のプリント。

【請求項44】 前記展開手段は、前のプリント処理が終了するまで変化させた画像の展開を延期することを特徴とする請求項41記載のプリント。

50 【請求項45】 前記予測手段は演算能力、メモリ量、バンド幅の少なくとも1つを予測し、前記変更手段はメ

メモリが足りなくなる場合により多くの演算能力を用いるような符号化表現に変更することを特徴とする請求項30記載のプリント方法。

【請求項46】 前記予測手段は演算能力、メモリ量、バンド幅の少なくとも1つを予測し、前記変更手段は演算能力やメモリ量が足りなくなる場合により多くのバンド幅を用いるような符号化表現に変更することを特徴とする請求項30記載のプリント方法。

【請求項47】 前記予測手段は演算能力、メモリ量、バンド幅の少なくとも1つを予測し、前記変更手段はプリント資源の範囲内で前記展開手段によってラスタリングされるような符号化表現に変更することを特徴とする請求項30記載のプリント方法。

【請求項48】 前記展開手段がプリント資源の範囲内で画像をラスタリングできるように、前記変更手段で符号化表現を変更することが不可能な場合には、前記変更手段は画像の質を変化させることを特徴とする請求項47記載のプリント方法。

【請求項49】 前記変更手段が画質を変化させる場合には、前記変更手段は画質の変化に先立って前記展開手段でもって前のページのラスタリング処理を行うことを特徴とする請求項48記載のプリント方法。

【請求項50】 解像度は、画素密度、カラー解像度、グレイレベル数の少なくとも1つを含むことを特徴とする請求項49記載のプリント方法。

【請求項51】 プリントメモリを監視して、プリントする物理ページに対応する前記プリントメモリの部位を特定する監視手段と、前記プリントメモリに画像データのバンドの符号化表現を記憶する符号化手段と、画像データの1ページの符号化処理が終了した時点で、画像データのバンドの符号化表現を前記プリントメモリのバンドラスタ部位に展開する展開手段と、バンドラスタ部位への展開処理が終了した時点で、前記プリントメモリのバンドラスタ部位中のラスタ画像データを一定速度でプリントエンジンに転送する転送手段とを備え、前記符号化手段は、前記監視手段によって監視されたページ位置で画像データの次のページを符号化することを特徴とするプリントメモリ・コントローラ。

【請求項52】 前記展開手段は符号化画像を変化した解像度でもとの位置に展開することを特徴とする請求項51記載のプリントメモリ・コントローラ。

【請求項53】 前記展開手段が符号化情報ページをもとの位置に展開する場合、前記展開手段が符号化画像を展開する以前に、前記転送手段がそれ以前のすべてのページを転送することを特徴とする請求項51記載のプリントメモリ・コントローラ。

【請求項54】 プリント処理の開始に先立って必要となる資源を予測し、予測結果に基づいてプリント処理を

適応的に変更するプリント方法。

【請求項55】 前記適応的にプリント処理を変更する行程は、プリントデータの符号化表現をラスタ画像に展開する行程を含み、前記必要となる資源の予測行程は符号化表現に基づいて行われることを特徴とする請求項54記載のプリント方法。

【請求項56】 前記適応的にプリント処理を変更する行程は、予測結果に基づいて画質を変化させることを特徴とする請求項54記載のプリント方法。

10 【請求項57】 プリント処理において必要となる資源を予測する予測行程と、該予測行程によって予測された資源が、プリント資源の範囲内であるかどうかを判断する判断行程と、前記判断行程によってプリント処理に必要な資源が足りないとは判断された場合に、プリント処理を変更する変更行程とを備えることを特徴とするプリント方法。

20 【請求項58】 プリント処理中に画像の符号化表現を形成する行程をさらに備え、符号化表現をフルラスタ画像に展開する間に前記予測行程が必要となる資源を予測することを特徴とする請求項57記載のプリント方法。

30 【請求項59】 プリント処理はプリント情報の各ページ中のバンドごとに行われ、前記予測行程は必要となる資源を各バンドごとに予測することを特徴とする請求項57記載のプリント方法。

40 【請求項60】 前記変更行程に基づくプリント処理の変更に応じて前記予測行程は必要となる資源を再予測し、前記変更行程は該予測行程によって予測された資源がプリント資源の範囲内に収まるまでプリント処理を繰り返して変更することを特徴とする請求項57記載のプリント方法。

【請求項61】 前記変更行程は前記プリントでプリントされる画像の質を変化させてプリント処理を変更することを特徴とする請求項57記載のプリント方法。

【請求項62】 解像度を変化させることを特徴とする請求項61記載のプリント方法。

【請求項63】 カラー解像度もしくは中間調レベル数を減少させることを特徴とする請求項61記載のプリント方法。

40 【請求項64】 一定の速度でプリント情報をスキャンしてプリントし、少なくとも一部のプリント情報を一定速度で出力することのできるビットマップメモリを用いる行程をさらに備えることを特徴とする請求項57記載のプリント方法。

【請求項65】 前記予測行程は、画像を前記ビットマップメモリに生成するのに必要となる演算能力、画像を記憶するのに必要となるメモリ量、画像をビットマップメモリに転送するのに必要となるバンド幅のうちの少なくとも1つを予測することを特徴とする請求項64記載のプリント方法。

50 【請求項66】 入力画像データをプリント画像の符号

化表現に翻訳する翻訳行程と、
プリント画像の符号化表現を記憶する記憶行程と、
記憶されている符号化表現をフルスタ画像に展開する
のに必要な資源を予測する予測行程と、
必要となる資源がプリント資源の範囲内であると予測さ
れた場合には、記憶されている符号化表現をラスト画像
に展開する展開行程と、

ラスト画像を一定速度でスキャンしてプリントするプリ
ント行程とを備えることを特徴とするプリント方法。

【請求項67】 前記予測行程によって符号化表現をフル
スタ画像に展開するのに必要な資源が足りないと判断
された場合には、プリント処理を変更する変更行程を
さらに備えることを特徴とする請求項66記載のプリン
ト方法。

【請求項68】 前記変更行程は、必要となるプリント
資源が異なるように符号化表現を別の表現に変更するこ
とを特徴とする請求項67記載のプリント方法。

【請求項69】 前記変更行程はプリント解像度を变化
させることを特徴とする請求項67記載のプリント方
法。

【請求項70】 前記変更行程はカラー解像度もしくは
中間調レベル数を減少させることを特徴とする請求項6
7記載のプリント方法。

【請求項71】 前記予測行程は、少なくとも符号化表
現をラスト画像に展開するのに必要な演算能力と、符号
化表現を記憶するのに必要なメモリ量とを予測するこ
とを特徴とする請求項66記載のプリント方法。

【請求項72】 前記翻訳行程によって翻訳される画像
データを入力する行程をさらに備えることを特徴とする
請求項66記載のプリント方法。

【請求項73】 画像データはページ記述言語で入力さ
れることを特徴とする請求項72記載のプリント方法。

【請求項74】 前記メモリ第1部位は符号化表現を
バンドごとに記憶し、前記メモリ第2部位は対応する
バンド情報のラスト画像を記憶することを特徴とする
請求項66記載のプリント方法。

【請求項75】 前記翻訳行程は入力画像データをバン
ドで翻訳することを特徴とする請求項74記載のプリン
ト方法。

【請求項76】 前記予測行程はプリント情報の各バン
ドごとに必要となる資源を予測することを特徴とする請
求項75記載のプリント方法。

【請求項77】 前記判断行程によって資源が足りないと
判断された場合には、前記展開行程はフルスタ画像を
前記メモリ第1部位と第2部位のものの位置に展開
することを特徴とする請求項76記載のプリント方法。

【請求項78】 前記翻訳行程は入力情報を符号化ラス
タ表現のバンドに翻訳し、前記判断行程が前記プリン
タエンジンの一定速度に対応する所定の時間間隔でバン
ド情報を展開するのに必要な演算能力が足りないと判断し

た場合に、該展開行程がバンドを前記メモリ第3部位
に前もってラストリングすることを特徴とする請求項6
6記載のプリント方法。

【請求項79】 前記予測行程によって符号化表現の方
が対応するラスト画像以上にメモリを必要とすると判断
された場合には、前記変更行程がプリント処理を変更す
ることを特徴とする請求項66記載のプリント方法。

【請求項80】 前記判断行程によって1つのバンド情
報に必要な資源が足りないと判断された場合には、それ
以外のバンドはそのまま、当該バンド情報のみを劣化
させることを特徴とする請求項66記載のプリント方
法。

【請求項81】 前記表現は、ビットマップオブジェ
クト、ラン長オブジェクト、台形オブジェクト、ボックス
オブジェクト、固定境界符号オブジェクトの少なくとも
1つから成ることを特徴とする請求項66記載のプリン
ト方法。

【請求項82】 前記展開行程は、アプリケーション機
能によってオブジェクトを背景に対して適用し、表現の
ラストリング処理を行う行程を含むことを特徴とする請
求項81記載のプリント方法。

【請求項83】 複数のバンドの各画像データを画像の
符号化表現に翻訳する翻訳行程と、

符号化表現を記憶する行程と、

記憶されている符号化表現に基づいて、符号化表現のバ
ンドをラスト画像に展開するのに必要な資源を予測する
予測行程と、

必要となる資源がプリント資源では足りないと予測され
た場合に、符号化表現を変更する変更行程と、

符号化表現のバンドをラスト画像データのバンドに展開
する展開行程と、

ラスト画像データのバンドを一定速度でプリントするプリ
ント行程とを備えることを特徴とするプリント方法。

【請求項84】 入力画像データをページ記述言語で入
力する行程をさらに備えることを特徴とする請求項83
記載のプリント方法。

【請求項85】 前記翻訳行程はページ記述言語で表わ
された画像データを、ビットマップオブジェクト、ラン
長オブジェクト、台形オブジェクト、ボックスオブジェ
クト、固定境界符号オブジェクトの少なくとも1つを含
む符号化表現に翻訳することを特徴とする請求項84記
載のプリント方法。

【請求項86】 前記展開行程は、アプリケーション処
理によってオブジェクトを背景に対して適用し、画像を
ラストリングする行程を備えることを特徴とする請求項
85記載のプリント方法。

【請求項87】 前記予測行程は、符号化表現を構成す
るプリミティブに基づいて必要となる資源を予測するこ
とを特徴とする請求項85記載のプリント方法。

【請求項88】 前記予測行程は画像の各バンドごとに

必要となる資源を予測することを特徴とする請求項83記載のプリント方法。

【請求項89】 前記予測行程は、演算能力、メモリ能力、バンド幅の少なくとも1つを予測することを特徴とする請求項83記載のプリント方法。

【請求項90】 前記変更行程は、演算能力の資源が過大である場合に、より多くのメモリを用いるような符号化表現に変更することを特徴とする請求項89記載のプリント方法。

【請求項91】 前記変更行程は、前記展開行程によって画像データのバンドを前もってラスティングすることを特徴とする請求項90記載のプリント方法。

【請求項92】 前記変更行程は画像の質を変化させることを特徴とする請求項90記載のプリント方法。

【請求項93】 前記変更行程は、必要となる資源がプリンタ資源以上となるバンドに対しては少なくとも前記展開行程によって画質を変化させることを特徴とする請求項92記載のプリント方法。

【請求項94】 前記変更行程は、前記展開行程によって全体の画像の質を変化させ、変化させた画像を前記符号化表現によってもとの位置に展開することを特徴とする請求項93記載のプリント方法。

【請求項95】 前記画質の劣化は、画素密度、カラー解像度、グレイレベル数の少なくとも1つに対して行われることを特徴とする請求項93記載のプリント方法。

【請求項96】 前記画質の劣化は、画素密度、カラー解像度、グレイレベル数の少なくとも1つに対して行われることを特徴とする請求項94記載のプリント方法。

【請求項97】 前記展開行程は、前のプリント処理が終了するまで変化させた画像の展開を延期することを特徴とする請求項94記載のプリント方法。

【請求項98】 前記予測行程は、演算能力、メモリ量、バンド幅の少なくとも1つを予測し、前記変更行程はメモリが足りなくなる場合により多くの演算能力を用いるような符号化表現に変更することを特徴とする請求項83記載のプリント方法。

【請求項99】 前記予測行程は、演算能力、メモリ量、バンド幅の少なくとも1つを予測し、前記変更行程は演算能力やメモリ量が足りなくなる場合により多くのバンド幅を用いるような符号化表現に変更することを特徴とする請求項83記載のプリント方法。

【請求項100】 前記予測行程は、演算能力、メモリ量、バンド幅の少なくとも1つを予測し、前記変更行程はプリンタ資源の範囲内で前記展開行程によってラスティングされるような符号化表現に変更することを特徴とする請求項83記載のプリント方法。

【請求項101】 前記展開行程がプリンタ資源の範囲内で画像をラスティングできるように、前記変更行程で符号化表現を変更することが不可能な場合には、前記変更行程は画像の質を劣化させることを特徴とする請求項

100記載のプリント方法。

【請求項102】 前記変更行程が画質を変化させる場合には、前記変更行程は画質の変化に先立って前記展開行程によって前のページのラスティング処理を行うことを特徴とする請求項101記載のプリント方法。

【請求項103】 解像度は、画素密度、カラー解像度、グレイレベル数の少なくとも1つを含むことを特徴とする請求項102記載のプリント方法。

【請求項104】 プリントメモリを監視して、プリントする物理ページに対応する前記プリントメモリの部位を特定する監視行程と、前記プリントメモリに画像データのバンドの符号化表現を記憶する記憶行程と、

画像データの1ページの符号化が終了した時点で、画像データのバンドの符号化表現を前記プリントメモリのバンドラスタ部位に展開する展開行程と、

バンドラスタ部位へのラスティングが終了した時点で、前記プリントメモリのバンドラスタ部位中のラスタ画像データを一定速度でプリンタエンジンに転送する転送行程とを備え、

前記符号化手段は、前記監視行程によって監視されたページ位置で画像データの次のページを符号化することを特徴とするプリントメモリの制御方法。

【請求項105】 前記展開行程は符号化画像を劣化解像度でもとの位置に展開することを特徴とする請求項104記載のプリントメモリの制御方法。

【請求項106】 前記展開行程が符号化情報ページをもとの位置に展開する場合、該展開行程が符号化画像を展開する以前に、前記転送行程がそれ以前のすべてのページを転送することを特徴とする請求項104記載のプリントメモリの制御方法。

【請求項107】 ラスタベースの文書のためのグラフィックス言語コマンドを生成する方法であって、装置に依存しないプリントコマンドを含むプリントコマンドを入力する行程と、

入力される前記プリントコマンドに基づいて、文書中のラスタフレームを記述するフレーム記述コマンドを生成する行程と、

入力される前記プリントコマンドに基づいて、ラスタフレームをレイアウトするための文書レイアウトコマンドを生成する行程と、

前記フレーム記述コマンドと文書レイアウトコマンドとをそれぞれ第1と第2のパイプラインインタフェースに出力する行程とを備えることを特徴とする方法。

【請求項108】 入力される前記プリントコマンドはページ記述言語から成り、該ページ記述言語をグラフィックス言語に変換する行程をさらに備えることを特徴とする請求項107記載の方法。

【請求項109】 フレーム記述中のフレームをラスタライズする行程をさらに備えることを特徴とする請求項

107記載の方法。

【請求項110】 前記文書レイアウトコマンドに基づいて、ラスタライズされたフレームをビットマップラスタに展開する行程をさらに備えることを特徴とする請求項109記載の方法。

【請求項111】 前記文書レイアウトコマンドは紙シートハンドリングのためのコマンドを備え、該紙シートハンドリングのコマンドに基づいてビットマップラスタを紙シート上にプリントする行程をさらに備えることを特徴とする請求項110記載の方法。

【請求項112】 前記フレーム記述コマンドはマスク生成コマンドと背景生成コマンドとを有するラスタベースのフレーム記述コマンドであり、マスク生成コマンドに応じて生成されるマスクを背景生成コマンドに応じて生成される背景にビットマップメモリ上で適用する行程をさらに備えることを特徴とする請求項107記載の方法。

【請求項113】 前記フレーム記述コマンドは装置依存のコマンドであることを特徴とする請求項112記載の方法。

【請求項114】 前記方法はホストCPUで実行されるプリンタドライバにより構成されることを特徴とする請求項107記載の方法。

【請求項115】 画素オブジェクトを生成するマスク生成手段と、
該マスク生成手段によって生成された画素オブジェクトを適用する背景を生成する背景生成手段と、
グラフィックス言語コマンドを解釈し、解釈したグラフィックス言語コマンドに応じて、前記マスク生成手段と背景生成手段のどちらか1つを起動する解釈手段と、
前記マスク生成手段によって生成された画素オブジェクトを、前記背景生成手段によって生成される背景にビットマップラスタメモリ上で適用するラスタ手段とを備えるグラフィックス言語処理装置。

【請求項116】 前記マスク生成手段はそれぞれが異なるマスクを生成する複数のマスク生成手段を備え、前記解釈手段は解釈したグラフィックス言語に基づいて該複数のマスク生成手段の1つを起動することを特徴とする請求項115記載のグラフィックス言語処理装置。

【請求項117】 前記複数のマスク生成手段の各々は、プリミティブ画素オブジェクトのコンパクト表現を生成するように構成されていることを特徴とする請求項第116項記載のグラフィックス言語処理装置。

【請求項118】 前記背景生成手段はそれぞれが異なる背景を生成する複数の背景生成手段を備え、前記解釈手段は解釈したグラフィックス言語に基づいて該複数の背景生成手段の1つを起動することを特徴とする請求項115記載のグラフィックス言語処理装置。

【請求項119】 前記ラスタ手段は、選択可能なアプリケーションに基づいて画素オブジェクトを背景に適

用するように構成されていることを特徴とする請求項115記載のグラフィックス言語処理装置。

【請求項120】 前記マスク生成手段によって生成された画素オブジェクトをクリッピングするクリップをさらに備えることを特徴とする請求項115記載のグラフィックス言語処理装置。

【請求項121】 ラスタ領域を逐次的に選択し、選択したラスタ領域に対して前記マスク生成手段と前記背景生成手段と前記ラスタ手段の各々を動作させるようなディスパッチャをさらに備えることを特徴とする請求項115記載のグラフィックス言語処理装置。

【請求項122】 選択した領域がラスタをバンドに分割した領域であることを特徴とする請求項121記載のグラフィックス言語処理装置。

【請求項123】 選択した領域がラスタを任意に分割した領域であることを特徴とする請求項121記載のグラフィックス言語処理装置。

【請求項124】 前記マスク生成手段と前記背景生成手段と前記ラスタ手段とはパイプライン処理構成になっており、パイプライン処理を調整するための調整手段をさらに備えることを特徴とする請求項115記載のグラフィックス言語処理装置。

【請求項125】 符号化されたプリントデータを入力する行程と、
符号化された前記プリントデータを第1の符号化表現に変換する行程と、
第1の符号化表現を第1の符号化表現とは異なる第2の符号化表現に変換する行程と、
第2の符号化表現からラスタ画像を形成する行程とを備えることを特徴とするプリント方法。

【請求項126】 前記符号化されたプリントデータはページ記述言語プリントデータであり、前記第1と第2の符号化表現はページ記述言語とラスタ画像の中間の複雑さを有することを特徴とする請求項125記載のプリント方法。

【請求項127】 前記第1と第2の符号化表現は複数の符号化プリミティブを有し、前記第2の符号化表現のすべての符号化プリミティブは少なくとも前記第1の符号化表現の符号化プリミティブのいくつかと同一であることを特徴とする請求項125記載のプリント方法。

【請求項128】 前記第2の符号化表現は、前記ラスタ行程において必要となるプリンタ資源を精度良く予測できるような正確なモデリング機能を有し、前記第1の符号化表現は前記モデリング機能を持たないことを特徴とする請求項125記載のプリント方法。

【請求項129】 プリンタ資源が不足すると予測された場合にプリンタ資源を再割り当てする行程をさらに備えることを特徴とする請求項128記載のプリント方法。

【請求項130】 ラスタベースの文書のためのグラフ

13

ィックス言語コマンドを生成する装置であって、装置に依存しないプリントコマンドを含むプリントコマンドを入力する入力手段と、

入力される前記プリントコマンドに基づいて、文書中のラストフレームを記述するフレーム記述コマンドを生成する第1の生成手段と、

入力される前記プリントコマンドに基づいて、ラストフレームをレイアウトするための文書レイアウトコマンドを生成する第2の生成手段と、

前記フレーム記述コマンドと文書レイアウトコマンドをそれぞれ第1と第2のパイプラインインタフェースに出力する出力手段とを備えることを特徴とする装置。

【請求項131】 入力される前記プリントコマンドはページ記述言語から成り、該ページ記述言語をグラフィックス言語に変換する変換手段をさらに備えることを特徴とする請求項130記載の装置。

【請求項132】 フレーム記述中のフレームをラストライズするラスト手段をさらに備えることを特徴とする請求項130記載の装置。

【請求項133】 前記文書レイアウトコマンドに基づいて、ラストライズされたフレームをビットマップラスタに展開する展開手段をさらに備えることを特徴とする請求項132記載の装置。

【請求項134】 前記文書レイアウトコマンドは紙シートハンドリングのためのコマンドを備え、該紙シートハンドリングのコマンドに基づいてビットマップラスタを紙シート上にプリントするプリント手段をさらに備えることを特徴とする請求項133記載の装置。

【請求項135】 前記フレーム記述コマンドはマスク生成コマンドと背景生成コマンドとを有するラストベースのフレーム記述コマンドであり、該マスク生成コマンドに応じて生成されるマスクを該背景生成コマンドに応じて生成される背景にビットマップメモリ上で適用する適用手段をさらに備えることを特徴とする請求項130記載の装置。

【請求項136】 前記フレーム記述コマンドは装置依存のコマンドであることを特徴とする請求項135記載の装置。

【請求項137】 前記装置はホストCPUで実行されるプリンタドライバにより構成されることを特徴とする請求項130記載の装置。

【請求項138】 グラフィックス言語を処理する方法であって、

画素オブジェクトを生成する行程と、

該画素オブジェクト生成行程によって生成された画素オブジェクトを適用する背景を生成する行程と、

グラフィックス言語コマンドを解釈し、解釈したグラフィックス言語コマンドに応じて前記2つの生成行程のどちらか1つを起動する行程と、

前記画素オブジェクト生成行程によって生成された画素

14

オブジェクトを、前記背景生成行程によって生成される背景にビットマップラスタメモリ上で適用する行程とを備える方法。

【請求項139】 前記画素オブジェクト生成行程は複数のそれぞれ異なるマスクを生成する行程を備え、前記解釈行程は解釈したグラフィックス言語に基づいて前記マスク生成行程で生成された該複数のマスクの1つを起動することを特徴とする請求項138記載の方法。

【請求項140】 前記マスク生成行程で生成された前記複数のマスクの各々は、プリミティブ画素オブジェクトのコンパクト表現を生成するように構成されていることを特徴とする請求項139記載の方法。

【請求項141】 前記背景生成行程は複数のそれぞれ異なる背景を生成する行程を備え、前記解釈する行程は解釈したグラフィックス言語に基づいて前記背景生成行程で生成された該複数の背景の1つを起動することを特徴とする請求項138記載の方法。

【請求項142】 前記適用行程は、選択可能なアプリケーションに基づいて画素オブジェクトを背景に適用することを特徴とする請求項138記載の方法。

【請求項143】 前記画素オブジェクト生成行程で生成された画素オブジェクトをクリッピングする行程をさらに備えることを特徴とする請求項138記載の方法。

【請求項144】 ラスタ領域を逐次的に選択し、選択したラスタ領域に対して前記画素オブジェクト生成行程と前記背景生成行程と前記適用行程の各々を動作させるような行程をさらに備えることを特徴とする請求項138記載の方法。

【請求項145】 選択した領域がラスタをバンドに分割した領域であることを特徴とする請求項144記載の方法。

【請求項146】 選択した領域がラスタを任意に分割した領域であることを特徴とする請求項144記載の方法。

【請求項147】 前記画素オブジェクト生成行程と前記背景生成行程と前記適用行程とはパイプライン処理構成になっており、パイプライン処理を調整するための行程をさらに備えることを特徴とする請求項138記載の方法。

【請求項148】 符号化されたプリントデータを入力する入力手段と、

符号化された前記プリントデータを第1の符号化表現に変換する第1の変換手段と、

該第1の符号化表現を該第1の符号化表現とは異なる第2の符号化表現に変換する第2の変換手段と、

該第2の符号化表現からラスタ画像を形成する形成手段とを備えることを特徴とするプリント装置。

【請求項149】 符号化された前記プリントデータはページ記述言語プリントデータであり、前記第1と第2の符号化表現はページ記述言語とラスタ画像の中間の複

15

16

雑さを有することを特徴とする請求項148記載の装置。

【請求項150】 前記第1と第2の符号化表現は複数の符号化プリミティブを有し、前記第2の符号化表現のすべての符号化プリミティブは少なくとも前記第1の符号化表現の符号化プリミティブのいくつかと同一であることを特徴とする請求項148記載の装置。

【請求項151】 前記第2の符号化表現は、前記ラスライズ行程において必要となるプリンタ資源を精度良く予測できるような正確なモデリング機能を有し、前記第1の符号化表現は該モデリング機能を持たないことを特徴とする請求項148記載の装置。

【請求項152】 プリンタ資源が不足すると予測された場合にプリンタ資源を再割り当てする再割り当て手段をさらに備えることを特徴とする請求項151記載の装置。

【請求項153】 画像を形成する装置であって、プリントコマンドを入力する通信インタフェースと、該プリントコマンドに基づいて、ラスタフレームを記述するフレーム記述コマンドとラスタフレームをレイアウトする文書レイアウトコマンドとを有するグラフィックス言語コマンドを生成する生成モジュールと、ラスタ画像情報を記憶するビットマップメモリと、前記フレーム記述コマンドに応じてラスタフレームを展開し、前記文書レイアウトコマンドに応じて前記ビットマップメモリに展開されたラスタフレームを記憶するグラフィックス言語プロセッサと、前記ビットマップメモリ中のラスタ画像情報に応じて画素ベースの画像を形成する画像形成手段とを備えることを特徴とする装置。

【請求項154】 前記画像形成手段は、ラスタ画像情報に応じてレーザビームを変調するドライバと、変調されるレーザビームを偏向して感光面上をスキャンする偏向手段とを備え、前記ドライバと前記偏向手段とを調整するシステムスケジューラをさらに備えることを特徴とする請求項153記載の装置。

【請求項155】 前記ビットマップメモリは複数のビットマップメモリに分割可能なビットマップラスタブールから構成され、空のビットマップメモリを前記グラフィックス言語プロセッサに転送し、描画されたビットマップメモリを前記画像形成手段に転送するシステムスケジューラをさらに備えることを特徴とする請求項153記載の装置。

【請求項156】 前記文書レイアウトコマンドに基づいてシートを前記画像形成手段に機械的に転送するシートハンドリング手段をさらに備えることを特徴とする請求項153記載の装置。

【請求項157】 画像形成後に、前記文書レイアウトコマンドに基づいてシートを処理する仕上げ手段をさらに備えることを特徴とする請求項156記載の装置。

【請求項158】 前記グラフィックス言語プロセッサはパイプラインプロセッサで構成されることを特徴とする請求項153記載の装置。

【請求項159】 前記グラフィックス言語プロセッサは、マスクを生成するマスク生成手段と、マスクをラスタフレームに適用する背景を生成する背景生成手段とを備え、該マスク生成手段と該背景生成手段とはグラフィックス言語コマンド列中のコマンドに応答することを特徴とする請求項153記載の装置。

【請求項160】 グラフィックス言語コマンド列を解析して、マスクコマンドを前記マスク生成手段に転送し、背景コマンドを前記背景生成手段に転送する解釈手段をさらに備えることを特徴とする請求項159記載の装置。

【請求項161】 選択可能なアプリケーションに応じて、前記マスク生成手段で生成されたマスクを、ラスタフレーム上の前記背景生成手段で生成された背景に適用してラスタフレームを展開するラスター手段をさらに備えることを特徴とする請求項160記載の装置。

【請求項162】 前記マスク生成手段と前記背景生成手段と前記ラスタリング手段は、パイプライン処理構造で構成されることを特徴とする請求項161記載の装置。

【請求項163】 パイプライン処理構造のパイプライン処理を調整するための調整手段をさらに備えることを特徴とする請求項162記載の装置。

【請求項164】 ラスタ領域を逐次的に選択し、選択したラスタ領域に対して前記マスク生成手段と前記背景生成手段と前記ラスタライズ手段の各々を動作させるようなディスパッチャ手段をさらに備えることを特徴とする請求項161記載の装置。

【請求項165】 選択した領域がラスタをバンドに分割した領域であることを特徴とする請求項164記載の装置。

【請求項166】 選択した領域がラスタを任意に分割した領域であることを特徴とする請求項164記載の装置。

【請求項167】 前記生成モジュールで生成されたグラフィックス言語コマンドは、グラフィックスオブジェクトを定義するコマンドと、オブジェクトをラスタフレームにどのように適用するのかを定義するコマンドとを有することを特徴とする請求項159記載の装置。

【請求項168】 前記グラフィックス言語プロセッサは、グラフィックス言語コマンドをグラフィックス言語コマンドとは異なる符号化コマンド列に変換する手段を備えることを特徴とする請求項153記載の装置。

【請求項169】 前記グラフィックス言語コマンドと符号化コマンド列は複数のコマンドプリミティブを有し、前記符号化コマンド列のすべてのコマンドプリミティブは少なくとも前記グラフィックス言語コマンドのコ

(10)

18

マンドプリミティブのいくつかが同一であることを特徴とする請求項168記載の装置。

【請求項170】 前記符号化コマンド列は必要となる画像形成資源を精度良く予測できるような正確なモデリング機能を有し、前記グラフィックス言語コマンドは該モデリング機能を持たないことを特徴とする請求項169記載の装置。

【請求項171】 画像形成資源が不足すると予測された場合に画像形成資源を再割り当てすることを特徴とする請求項170記載の装置。

【請求項172】 画像を形成する方法であって、

プリントコマンドを入力する行程と、
該プリントコマンドに基づいて、ラスタフレームを記述するフレーム記述コマンドとラスタフレームをレイアウトする文書レイアウトコマンドとを有するグラフィックス言語コマンドを生成する行程と、
ラスタ画像情報をビットマップメモリに記憶する行程と、

前記フレーム記述コマンドに応じてラスタフレームを展開し、前記文書レイアウトコマンドに応じて前記ビットマップメモリに展開されたラスタフレームを記憶する行程と、
前記ビットマップメモリ中のラスタ画像情報に応じて画素ベースの画像を形成する行程とを備えることを特徴とする方法。

【請求項173】 前記画像形成行程は、ラスタ画像情報に応じてレーザビームを変調する行程と、変調される該レーザビームを偏向して感光面上をスキャンする行程とを備え、前記方法は前記変調行程と前記偏向行程とを調整する行程をさらに備えることを特徴とする請求項172記載の方法。

【請求項174】 前記ビットマップメモリは複数のビットマップメモリに分割可能なビットマップラスタアップロードメモリに分割可能なビットマップラスタアップロードメモリから構成され、空のビットマップメモリを前記グラフィックス言語プロセッサに転送し、充填されたビットマップメモリを前記画像形成行程に転送する行程をさらに備えることを特徴とする請求項172記載の方法。

【請求項175】 前記文書レイアウトコマンドに基づいてシートを機械的に転送する行程をさらに備えることを特徴とする請求項172記載の方法。

【請求項176】 画像形成後に、前記文書レイアウトコマンドに基づいてシートを処理する行程をさらに備えることを特徴とする請求項175記載の方法。

【請求項177】 前記展開行程はパイプラインプロセッサで構成されることを特徴とする請求項172記載の方法。

【請求項178】 前記展開行程はマスクを生成する行程と、マスクを適用する背景をラスタフレームに生成する行程とを備え、該マスク生成行程と背景生成行程はグラフィックス言語コマンド列中のコマンドに応答するこ

とを特徴とする請求項172記載の方法。
【請求項179】 グラフィックス言語コマンド列を解析して、マスクコマンドを前記マスク生成行程に転送し、背景コマンドを前記背景生成行程に転送する行程をさらに備えることを特徴とする請求項178記載の方法。

【請求項180】 選択可能なアプリケーションに応じ、前記マスク生成行程で生成されたマスクを、ラスタフレーム上の前記背景生成行程で生成された背景に適用してラスタフレームをレンダリングする行程をさらに備えることを特徴とする請求項179記載の方法。

【請求項181】 前記マスク生成行程と前記背景生成行程と前記ラスタフレーム・レンダリング行程は、パイプライン処理構造で構成されることを特徴とする請求項180記載の方法。

【請求項182】 パイプライン処理構造のパイプライン処理を調整する行程をさらに備えることを特徴とする請求項181記載の方法。

【請求項183】 ラスタ領域を逐次的に選択し、選択したラスタ領域に対して前記マスク生成行程と前記背景生成行程と前記ラスタフレーム・レンダリング行程の各々を動作させる行程をさらに備えることを特徴とする請求項180記載の方法。

【請求項184】 選択した領域がラスタをバンドに分割した領域であることを特徴とする請求項183記載の方法。

【請求項185】 選択した領域がラスタを任意に分割した領域であることを特徴とする請求項183記載の方法。

【請求項186】 前記背景生成行程で生成されたグラフィックス言語コマンドは、グラフィックスオブジェクトを定義するコマンドと、オブジェクトをラスタフレームにどのように適用するのかを定義するコマンドとを有することを特徴とする請求項178記載の方法。

【請求項187】 前記レンダリング行程は、グラフィックス言語コマンドをグラフィックス言語コマンドとは異なる符号化コマンド列に変換する手段を備えることを特徴とする請求項172記載の方法。

【請求項188】 前記レンダリング行程と符号化コマンド列は複数のコマンドプリミティブを有し、前記符号化コマンド列のすべてのコマンドプリミティブは少なくとも前記グラフィックス言語コマンドのコマンドプリミティブのいくつかが同一であることを特徴とする請求項187記載の方法。

【請求項189】 前記符号化コマンド列は必要となる画像形成資源を精度良く予測できるような正確なモデリング機能を有し、グラフィックス言語コマンドは該モデリング機能を持たないことを特徴とする請求項188記載の方法。

【請求項190】 画像形成資源が不足すると予測され

(11)

20

19

た場合に画像形成資源を再割り当てすることを特徴とする請求項189記載の方法。
【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、グラフィックス言語列に応じて画像を処理する方法ならびに装置に関するものである。

【0002】

【従来の技術】近年、選択的に画素をページ上にプリントして画像を形成するレーザビームプリンタなどのプリント装置が、柔軟性、プリント速度、低コストなどの観点から広く使われ始めている。このようなプリント装置はページをスキャンして画像をプリントするものであり、スキャンライン上の個々の画素を選択的にプリントする。1つのスキャンラインなどのプリンタではプリントして（レーザビームプリンタなどのプリンタではプリントして）（レーザビームプリンタなどのプリンタでは）次のスキャンラインの処理を行う。ページ上のすべてのスキャンラインからプリントされる画素を集めると、所望の画像が得られる。

【0003】画素をページにプリントするかどうか、スキャン処理の間に画素をどこにプリントするかは、記憶したプリント画像のラスタ画像に基づいてプリンタ制御部が決定する。ここで、ラスタはプリント画像中の各画素をコンピュータ表現したものであり、制御部中のビットマップメモリに記憶される。ビットマップメモリは、ページ上のプリントされる画素とビットマップメモリ中のメモリ位置との間に1対1の関係がある。

【0004】すべてのプリント装置は利用できる資源の範囲内で処理を進めなければならない。例えば、制御部は一般に中央処理ユニット（「CPU」）をなすマイクロプロセッサであるが、制御部の処理能力は限られており、ある時間間隔で扱える画像データなどのデータ量は限られる。また、制御部のビットマップメモリなどのメモリへのアクセスも制限される。さらに、外部メモリ装置などの他の装置を備えるプリントの介してアクセスするが、外部メモリ装置とは通信ラインを介してアクセスするため、バンド幅（データ転送速度）の制限が生じる。

【0005】従来のプリント装置では、利用可能な有限の資源を非効率的に用いている。例えば、図11はプリンタにおける代表的なCPU資源の利用効率を示している。図11に示されるように、CPU資源の利用効率はピークのCPU利用率と比較的低いCPU利用率との間で大幅に変動する。ピークのCPU利用率はラスタ画像をビットマップメモリに生成する際のページプリントの間に生じ、低CPU利用率はページがプリントされているときに生じる。

【0006】このような効率の悪さが生じるのは、画像を適切にプリントすることを保証するために、スキャン処理で画素を形成するようなレーザビームプリンタなど

のプリント装置は、プリント処理に先立ってプリント画像のフルラスタ画像をビットマップメモリに生成する必要があるためである。これは、プリント装置がページをプリントするときに一定の速度でスキャンしてプリントすることによる。すなわち、ページのプリントが始まるのと、ページのプリントが終わるまでプリント処理は一定の速度で行われる。したがって、プリントエンジンにおいてプリントすべきデータが欠如（データアンダーフローと呼ばれる状況）しないように、プリント処理に先立ってプリントすべき全体の画像をビットマップメモリ中のラスタ画像に変換する。CPUはラスタ画像をビットマップメモリに生成する処理を受け持ち、この処理のときにCPU利用率がピークとなる。これに対して、実際に画像をプリントしているときは、CPUはビットマップメモリを監視して、プリンタエンジンがプリント処理を行う速度でラスタ画像がプリンタエンジンに供給されているかどうかをチェックするだけでよい。このような比較的単純な処理のときには、CPU利用率が低くなる。

【0007】一方、図11にさらに示されているように、CPU利用率が最大であるときに、メモリ利用率は最低になる。ビットマップメモリは画像のプリント処理の直前でフルになるためである。上述のように、これはCPU利用率が最低である時点に対応する。単純なプリントシステムでは、このような非効率性は無視できる。例えば、単純な白黒画像を300ドット/インチ（dpi）などの低解像度で8.1/2 × 11インチの標準紙シートにプリントするには、およそ1メガバイト（Mb）のビットマップメモリが必要である。この際、1Mbのメモリの値段は安いので、メモリの大部分が長い時間使われなくても大きな問題ではない。ビットマップメモリがダブルバッファメモリ、すなわち2Mbメモリであっても同様である。なお、ダブルバッファリングは、CPUが次のページの画像データをラスタ画像として1つのバッファに生成する間に、他のバッファから現在のページの画像データをプリントすることができるといった好ましい性質を有する。すなわち、ダブルバッファリングによれば、ラスタ画像データ列を連続的にプリントエンジンに転送することが可能で、コストをあまり上げることなくページ内スルーアップを上げることが可能となる。

【0008】これに対して、解像度が増大すれば、またより多くの処理能力（例えば、中間色やカラー画像など）が望まなければ、このような非効率性を無視することはできなくなる。例えば、600dpiの解像度では、フルラスタ画像を記憶するのに300dpiの4倍ものメモリが必要となる。中間調やカラー画像を対象とするときには、これ以上のメモリがさらに必要となる。例えば、中間調画像において各画素ごとに16階調とすれば画素ごとに4ビット必要になる。画素ごとの4ビットに600dpiのラスタ画像を記憶するのに必要な4

Mbを掛けると、中間調ラスタ画像を記憶するには16 Mb必要となることがわかる。カラー画像を対象とするときには、これ以上のメモリが必要である。例えば、各色ごとに8ビットで画素ごとに4色で符号化するときには、各画素32ビット必要となる。32ビットと600 dpiのラスタ画像の4 Mbとを掛けると、フルカラーの600 dpiラスタ画像を記憶するには128 Mbのメモリが必要になる。このように大量のメモリの値段は商用の製品としては許容できる範囲ではない。また、ページ内の遅延を減少させるためにダブルバッファリングを用いるときには、問題はさらに深刻になる。

【0009】このような問題に対して、ディスクドライブなどのオフラインメモリ装置を他のメモリ空間として用いて、通信リンクを介してプリンタ制御部と接続するというような仮想メモリを利用して、物理メモリの容量を仮想的に増加させることが考えられている。しかし、プリンタ装置において仮想メモリを用いるのは効率上あまり好ましくない。データのアンダーフローを生じないように画素データを適切な時点でプリンタエンジンに転送するには、通信リンクのバンド幅が十分広くなければならず、この点での保証が得られないためである。

【0010】更に、近年、レーザビームプリンタ、インクジェットプリンタ、バブルジェットプリンタなど、高品質なラスタベースの装置が広く用いられ始めている。これらの装置では、ホストコンピュータから画像データを入力し、ビットマップメモリにラスタ画像を生成してから、プリント処理が行われる。図1は、ビットマップメモリ中の各画素を形成して、品質（色、グレイレベル、中間調など）を各画素に付与するラスタライズ処理を示した図である。図1に示されるように、ビットマップラスタ100はビットマップラスタ中の領域を定義するフレームFに分割される。また、フレーム中の各画素101ごとに、マスクMと背景Bが定義されている。ここで、マスクMは、画素をビットマップメモリに展開すべきかどうかを定義すると同時に、画素を展開すべき位置を定義するものである。また、背景Bは、画素101を展開すべき品質を定義するものである。例えば、背景Bには画素に割り当てられる色や、中間調密度が指定されている。なお、クリッピング領域を定義して、マスクの実効領域を指定して、マスクをより適確に制御することも可能である。画素101のビットマップメモリへのラスタライズ処理は、フレーム情報、背景情報、クリッピング領域情報、マスク情報（すべて図1に示されている）の論理和をとることで行われる。

【0011】ラスタプリント装置が種々開発されるにつれて、より洗練された手法でラスタ画像の形成が行われるようになってきた。すなわち、フルラスタ画像をプリンタに転送するのは実際的ではないため、ページ記述言語（「PDL」）のような記述言語が用いられるようになってきた。ページ記述言語は、各文書ページ上のテキ

スト、グラフィックス、スキャン画像などの各要素の性質をすべて記述するコマンド列から構成される。また、PDLは、文書のレイアウトやシートハンドリング（ページ順揃えなど）を記述する情報をも含む。ホストコンピュータがページ記述言語をプリント装置に転送すると、プリント装置がページ記述言語を解釈してラスタ画像に展開する。

【0012】図2は、このような処理を示す図である。図2は、一端でページ記述言語を入力し、他端でビットマップラスタ画像を生成するプリント装置中のモジュールを示している。図2に示されるように、通信インタフェースモジュール102はPDLコマンドを入力し、PDLインタプリタ104に転送する。PDLインタプリタ104は、インタプリタで処理したPDLコマンドに応じて、PDLコマンドをグラフィックスレイヤ105、フォントモジュール106、記憶プール107に転送する。ラスタライザ109は、グラフィックスレイヤ105、フォントモジュール106、記憶プール107からの出力を受け取って、ビットマップメモリにラスタ画像を生成する。すなわち、ラスタライザ109は、ビットマップラスタプール110の空ラスタを受け取り、画像を描画した後描画されたラスタをラスタプール110に出力する。充填されたラスタ画像はプリント装置によって出力されるが、この間にラスタライザ109はさらに次のページに対応する空ラスタに描画を行う。

【0013】PDLコマンドのようなプリンタコマンドは一般に装置に依存しない。すなわち、PDLコマンドを入力するプリント装置に関わらず、同一のPDLコマンドからは同一の文書が出力される。このような特徴は利点となりうるが、一方、各プリンタはそれぞれ個々の特徴を有しており、装置に依存しないPDLコマンドに対応するラスタ画像に変換するPDLインタプリタ104をプリンタごとに個別に用意しなければならないという問題点もある。

【0014】

【発明が解決しようとする課題】本発明の目的は、プリンタ内の資源をより効率的に利用することで、利用可能な物理的な資源を増加させることなくプリンタの性能を向上させることである。本発明の他の目的は、装置に依存しないPDLコマンドに対応するラスタ画像に変換するPDLインタプリタをプリンタごとに個別に用意する必要をなくし、汎用性を高めることである。

【0015】

【課題を解決するための手段】1つの態様において、本発明は、画像を符号化されたコンパクトな表現で表わし、その後でフルラスタ画像に展開するものである。この際の表現としては、フルラスタ画像を展開するのに必要な資源を精度良く予測することが可能な表現を用いる。また、予測処理は資源の利用率を予測するためのもので、プリント処理に先立って、プリンタがプリント処

23

理を行うのに十分な資源を有しているかどうかを判断する。ここで、資源が十分にあると判断された場合には、プリント処理が実行される。一方、資源が足りない場合には、資源の再割り当てを行ってからプリント処理を実行する。

【0016】他の態様として、本発明は、装置に依存しないプリントコマンドに基づいてグラフィックス言語を用いた文書記述コマンドを生成する。文書中のラスタフレームに対するフレームの記述は、ラスタフレームをページラスタ中にどのように配置するかを指定する文書レイアウトコマンドと同様に、プリントコマンドから生成される。フレームの記述は第1のパイプラインインタフェースから出力され、文書レイアウトコマンドは第2のパイプラインインタフェースから出力される。文書レイアウトコマンドには、フレームを紙シート上にどのようにマッピングするかについての情報や、とじこみ命令やページ順揃え命令などの文書仕上げ命令に関する情報なども含まれる。フレーム記述はラスタベースの記述であり、一般に装置依存型の記述である。本発明の実施例では、レーザビームプリンタ、インクジェット、バブル

ジェットプリンタなどのラスタベースのプリンタあるいは表示装置への適用であるが、ホストコンピュータ中のプリンタドライバへ適用して、ホストコンピュータ処理の一部として実行することもできる。

【0017】さらに他の態様として、本発明はグラフィックス言語プロセッサであって、画素マスクオブジェクトをフレームに生成するマスク生成部を備える。また、背景生成部を備え、オブジェクトが展開される背景を生成する。解釈部はプロセッサに入力したグラフィックス言語コマンド列を解釈し、グラフィックス言語に応じてマスク生成部と背景生成部とを起動する。ラスタライザ部は、マスクオブジェクトをビットマップラスタメモリ上で背景に対し展開する。マスク生成部は複数の異なるマスク生成部を備えており、これらの複数のマスク生成部はパイプライン構造で構成されて、グラフィックス言語に応じて解釈部によって起動される。同様に、背景生成部はそれぞれ異なる種類の背景を生成する複数の背景生成部を備えており、各背景生成部はパイプライン構造で構成されて、適当なグラフィックス言語に応じて解釈部によって起動される。調整部は構造中のパイプラインを調整するために設けられている。マスク生成部において生成されたマスクオブジェクトは、画素オブジェクトのコンパクト表現であるプリミティブオブジェクトのこともあり、またビットマップ画像のセグメントのこともある。また、マスク生成部の出力をクリッピングするためにクリッパを設けることも可能である。同様に、いくつかの実施例では、ディスパッチャを備えてラスタ領域を逐次的に選択し、マスク生成部、背景生成部、ラスタライザ部が選択されたラスタ領域それぞれに対して動作するように構成することもできる。例えば、ディスパ

24

ッチャを用いることで、バンドに分割したラスタを、マスク生成部、背景生成部、ラスタライザ部が逐次的に選択して処理するような構成が可能となる。

【0018】さらに他の態様として、本発明は、PDLプリントコマンドなどの符号化プリントデータを入力するステップと、符号化プリントデータを第1の符号化表現に変換するステップと、第1の符号化表現を第1の符号化表現とは異なる第2の符号化表現に変換するステップと、第2の符号化表現からラスタ画像を形成するステップとを備えるプリント方法である。第1と第2の符号化表現は共通の符号プリミティブを有し、第2の符号化表現は、表現をラスタライズする際に必要なプリンタ資源を正確に表わして、プリンタ資源が不足している場合にはラスタライズ処理を変更することができるような正確なモデリング機能を有する。

【0019】

【実施例】

<第1実施例>グラフィックス言語(GL)は、ページ中のフレームに対するグラフィックス記述と、フレームをページ中のどこに配置するかを記述する文書レイアウト情報とに分けることができ、グラフィックス記述と文書レイアウト情報とは別々のパイプライン化されたインタフェースから出力される。そこには、グラフィックス言語プロセッサが備えられ、グラフィックス言語列を解釈する。プロセッサはグラフィックス言語列をマスク情報とマスクを展開する背景とに分け、マスクと背景情報に別々に作用してフレームラスタを生成し、文書レイアウト情報を用いてフレームラスタをページにマッピングする。

【0020】図3は、本実施例のレーザビームプリンタ装置の構成を示す図である。図3に示されるように、レーザビームプリンタは、プリンタ内のさまざまな機構部や電子部を制御する制御部11を備える。具体的には、制御部11は不図示のホストコンピュータから通信ポート12を介して画像データを受信し、画像データを処理してレーザドライバ14を駆動するレーザ変調信号を生成する。ここで、通信ポート12から入力される画像データは、プリント画像のフルラスタ画像形式ではなく、ページ上のオブジェクトやオブジェクトの位置のみを指定するページ記述言語(「PDL」)のようなコンパクトなデータ形式であることが好ましいが、これ以外の画像データ形式であっても良い。制御部11で画像信号が生成されると、レーザドライバ14はレーザ15を駆動してレーザビーム16を放射する。回転ポリゴンミラー17はミラー19を介してレーザビームを偏向しながら感光ドラム20上をスキャンする。感光ドラム20は矢印Aの方向に回転し、帯電部21を通過する。スキャンされる偏向レーザビームは帯電された感光ドラム上に潜像を形成し、現像部22で潜像はトナー画像として現像される。

25

【0021】ドラム20表面をスキャンすると、制御部11はピックアップローラ25を駆動してトレー24中の最も上の紙シートを引き込み、ローラ26と27によって画像通路29を通して感光ドラム20上に運ばれる。ドラム20上のトナーで現像された潜像は転移帯電部30でこのシートに転移され、ローラ31と32で固着装置34に運ばれる。固着装置34において、熱と圧力を合せ加えてトナー画像がシートに永久的に固着されると、ローラ35によってシートは出力トレイ36に出力される。

【0022】図4は制御部11の構成を示す図である。図4中の通信インタフェース202とPDLインタプリタ204は、図2の通信インタフェースとPDLインタプリタと本質的に同一であるため、ここでの説明を省略する。図4に示されるように、ページ記述言語はPDLインタプリタ204で処理されてグラフィックス言語生成モジュール205に渡される。グラフィックス言語生成モジュール205では、PDLを本実施例のグラフィックス言語文書に変換する。以下、グラフィックス言語をGLとも記載する。具体的には、グラフィックス言語生成モジュール205は、PDL言語に基づいて、グラフィックス言語文書の2つの要素を生成する。まず、グラフィックス言語生成モジュール205は、グラフィックス言語のフレーム記述コマンド206を生成する。フレーム記述コマンド206は、アトリビュタスタ処理列としてラスタフレームを記述するものである。すなわち、フレーム記述コマンドでは、オブジェクトや、オブジェクトでラスタフレーム上をどのように塗るかといったルールが定義される。

【0023】例えば、フレーム記述コマンド206は、ボックス型画素をカラー背景画像上に配置し、処理画素をビットマップラスタメモリの所定の位置に入力するように、グラフィックス言語プロセッサに指示する。あるいは、フレームコマンドは、第1フレームに会社のレターヘッドを、第2フレームに手紙のテキスト画像を、第3フレームに署名の画像を形成するというように、グラフィックス言語プロセッサに指示する。フレーム記述コマンド206に関しては、以下でさらに詳細に説明する。

【0024】図4にさらに示されるように、グラフィックス言語生成モジュール205の第2の出力は、文書レイアウト情報207である。文書レイアウト情報207は、フレーム記述コマンド206で指定されたフレームで文書ページに対応するビットマップラスタ上をどのように塗るかを記述したものである。具体的には、文書レイアウト情報207は、フレームのラスタメモリへのマッピングの仕方に関する情報を含む。例えば、文書レイアウトコマンドは、複数のフレームを1ページのビットマップラスタ中の異なる位置（それぞれが重なっている場合もある）にマッピングするようにグラフィックス言

26

語プロセッサに指示する。また、プリント可能な文書の場合には、文書レイアウト情報は両面プリントなどのような特別なプリント命令を含むことがあり、とじ命令、ページ順揃え命令などのような紙シートハンドリング命令や仕上げ命令などを含むこともある。

【0025】グラフィックス言語コマンド206と文書レイアウト情報207とは、インタフェース208を介してパイプライン出力される。ここで、インタフェース208は、ホストCPUとプリントエンジン間のインタフェースのような個別の装置間の物理的インタフェースであっても良い。例えば、通信インタフェース202、PDLインタプリタ204、グラフィックス言語生成モジュール205をホストCPU中のプリントドライバとして構成することも可能である。このような場合、ホストコンピュータで生成されたページ記述言語コマンドはグラフィックス言語コマンド206と文書レイアウト情報207にまで変換されて、パイプラインインタフェースを介してプリンタエンジンに直接出力される。しかし、ここでは、インタフェース208を制御部11中の2つのモジュールを接続するインタフェースとして実施例を説明する。この際、モジュールはソフト的に構成されていても、ハード的に構成されていても良い。

【0026】グラフィックス言語コマンド206はグラフィックス言語プロセッサ209に入力され、ビットマップラスタプール210からの空のラスタにグラフィックス言語コマンドを展開する。ここで描画されたラスタはプリンタエンジン・ビデオインタフェース212に転送され、プリンタのレーザドライバ14に入力される。

【0027】文書レイアウト情報207はシステムスケジューラ214に入力され、さまざまなモジュールの処理を調整する。具体的には、システムスケジューラ214は、ビットマップラスタプール210からの空ラスタをグラフィックス言語プロセッサ209に渡し、画素画像のラスタへの展開処理の開始をグラフィックス言語プロセッサ209に指示する。グラフィックス言語プロセッサ209がラスタ画像への展開処理を終了すると、システムスケジューラ214は、展開されたラスタ画像に基づいてデータをレーザドライバ14に転送するようにプリンタエンジン・ビデオインタフェース212に指示する。レーザビームを変調すると同時に、システムスケジューラ214は、シート送り、ドラム回転、ローラ固定などのプリンタ10の機構動作をプリンタエンジンモジュール215に指示する。さらに、システムスケジューラ214は、文書レイアウト情報207に基づいて、両面プリント、紙出力、とじこみ、ページ順揃え、これら以外の文書仕上げ情報などを指示する。

【0028】図5はグラフィックス言語プロセッサ209の構成を詳細に示す図である。図示されているように、プロセッサ209は、グラフィックス言語コマンド列を分解し、グラフィックス言語コマンドを解釈するグ

27

グラフィックス言語インタプリタ220を備える。具体的には、グラフィックス言語インタプリタ220は、グラフィックス言語オブジェクトがグラフィックス言語コマンド列中に出てきたときには、それを記憶する。また、インタプリタ220は、マスクアプリケーションコマンドや背景生成コマンドがグラフィックス言語コマンド列中に出てきたときには、それらをそれぞれマスク生成部221と背景生成部222とに転送する。マスク生成部221は図1のマスクMなどのマスク情報を生成し、背景生成部222は図1の背景Bなどの背景情報を生成する。マスク生成部221と背景生成部222とも、GLオブジェクト記憶部224に蓄えられているグラフィックス言語オブジェクト上で処理を行う。

【0029】マスク生成部221において生成されたマスクはクリッパ225に転送され、必要であればGLコマンドに基づいてマスクのクリッピングを行う。クリッパ225もまた、GLオブジェクト記憶部224に蓄えられているグラフィックス言語オブジェクト上で処理を行う。クリッパ225と背景生成部222との出力は、ラスタOPマシーン226に入力される。ラスタOPマシーン226は、マスク生成部221により生成され、クリッパ225でクリッピングされたマスクを、背景生成部222で生成された背景上に適用する。この際、ラスタOPマシンは、GL状態メモリ227で指定されるアプリケーション機能に対応してマスクを適用する。例えば、図6に示されるように、不透明アプリケーション機能が指定される。すなわち、ラスタOPマシンのマスク生成部221で生成されクリッパ225でクリッピングされたマスクを、背景生成部222で生成された背景面に適用する際、もとのラスタ中の情報を上書きして、処理後のラスタを生成する。これ以外のアプリケーションを指定することも可能であり、例えば透明アプリケーションや付加アプリケーションなども可能である。

【0030】ラスタOPマシンの作用するラスタは、システムスケジューラ214に応じて、ビットマップラスタプール229から入力される。文書レイアウト情報207で述べたように、すべてのアプリケーションを終了すると、ラスタOPマシンは充填ラスタをビットマップラスタプール229に転送し、描画されたラスタはプリンタ10でプリントされる。

【0031】図7は、マスク生成部221と背景生成部222との構成、ならびにラスタOPマシーン226のアプリケーション処理を概念的に詳細に示す図である。図7では、以下の説明を簡潔にするために図5中のいくつかの要素を省略している。図7に示されるように、マスク生成部221は、特定の画素プリミティブオブジェクトを生成するように設計されたマスク生成部を複数備える。すなわち、マスク生成部221は、ビットマップマスク生成部221a、台形マスク生成部221b、ライン生成部221c、ランレングスマスク生成部221

28

d、ボックスマスク生成部221e、他のコンパクト表現を生成する生成部221fを備える。具体的には、ビットマップマスク生成部221aは、GLインタプリタ220からのGLコマンドに応じてビットマップマスクを生成する。ここで、ビットマップマスクは画素ビットごとにマスクを表現したものである。

【0032】台形マスク生成部221bは、GLインタプリタ220からのGLコマンドに応じて台形型のマスクを生成する。ここで、台形マスクはラスタメモリ中の台形オブジェクトを定義する。ライン生成部221cは、GLインタプリタ220からのGLコマンドに応じて指定された幅の線を生成する。ランレングスマスク生成部221dは、マスク内の画素のランレングスに応じてランレングスマスクを符号化する。ボックスマスク生成部221eは、GLインタプリタ220からのコマンドに応じて指定された幅と高さのボックス型のマスクを生成する。

【0033】なお、これら以外のコンパクト表現も可能であり、ここに示した表現のみに限定されるわけではない。マスク生成部221において生成されるマスクのうちの代表的な種類を示しただけである。背景生成部222は、カラー背景生成部222a、タイル背景生成部222b、画像背景生成部222cを備える。具体的には、カラー背景生成部222aは、GLインタプリタ220からのGLコマンドに応じて、指定されたカラーで一様なカラー背景を生成する。タイル背景生成部222bは、GLインタプリタ220で指定されたタイルを繰り返し生成する。すなわち、GLインタプリタで指定された領域の背景全体に、タイルパターンを繰り返し生成する。画像背景生成部222cは、GLインタプリタ220で指定された画素ごとのフォーマットで、画像背景を生成する。

【0034】なお、これら以外の背景生成部も可能であり、ここに示した生成部のみに限定されるわけではない。背景生成部222に含まれる背景生成部のうちの代表的な種類を示しただけである。図7には、ラスタOPマシーン226がグラフィックス言語コマンドに応じて、現在のラスタ230を更新して、新たなラスタ231を生成する様子も示されている。

【0035】図7は、GLインタプリタ220がグラフィックス言語コマンド「apply clipped box(W,H) at<x,y> with image background;<x,y>にあるクリッピングボックス(W,H)を画像背景に適用する」を入力した状態である。(なお、グラフィックス言語コマンドのリストは付録に添付されている)。GLインタプリタ220は、グラフィックス言語コマンドの前半部位、すなわち「apply clipped box(W,H)」に応じてボックスマスク生成部221eを起動する。ボックスマスク生成部221eは、マスク生成部221の出力に示されているように、幅W、高さHのボックス型のマスクを生成す

29

る。そこで、クリッパ225は、メモリ224に蓄えられているグラフィックス言語オブジェクトを切り出して、クリッパの隣に図示しているようなクリッピングされた形を生成する。そして、生成されたクリッピングボックスは、ラストOPマシン226に入力される。

【0036】また、GLEインタプリタ220は、グラフィックス言語コマンドの後半部位、すなわち「with image background」に応じて、画像背景生成部222cを起動する。画像背景生成部222cは、メモリ224に蓄えられているグラフィックス言語オブジェクトを切り出して、画像背景をラストOPマシン226に入力する。なお、この際、画像背景はカラーであっても良い。

【0037】クリッパ225と背景生成部222とから情報が入力されると、ラストOPマシン226は状態メモリ227で指定されるアプリケーション機能に基づいて、クリッピングされたマスクを背景に適用する。ここでは、アプリケーション機能は「不透明」と指定されており、ラストOPマシン226は背景の現在のラストの位置<x, y>をマスクされクリッピングされた領域に適用して、231に示されるような更新ラストを得る。以上の処理を各フレームのグラフィックス言語コマンド列のコマンドごとに実行し、その後で文書レイアウト情報207を用いて全体のビットマップラストにフレームをどのように配置するかを決定する。

【0038】図7では、図8のような逐次的な処理として説明を行った。すなわち、ステップ1で、マスク生成部221はマスクを生成して、クリッパ225に入力する。ステップ2において、クリッパ225はマスクを適当にクリッピングし、ラストOPマシン226に入力する。ステップ3で、背景生成部222は背景をラストOPマシン226に入力し、ステップ4で、ラストOPマシン226がマスクを背景に適用し、ビットマップラストメモリにラスト画像として書き込む。このようにグラフィックス言語プロセッサを構成することができるが、これ以外の構成法も考えることができる。

【0039】すなわち、図9に示すようにディスパッチャ330を設け、マスク生成部321、クリッパ325、背景生成部322、ラストOPマシン326の動作を制御することもできる。具体的には、ディスパッチャ330は領域330aなどのラスト画像中の特定領域で動作するように各モジュールに指示をする。このような領域としてはビットマップメモリ中の分割された領域バンドであることが多いが、必ずしもバンドである必要はなく、画像中の任意の領域をとることができる。ディスパッチャ330は、ビットマップメモリ中の全ページの展開をし終えるまで、例えばバンドごとに全体領域の処理を順々に進める。

【0040】すなわち、このようなディスパッチャ330によれば、マスク生成部321、背景生成部322、クリッパ325、ラストOPマシン326は、同時に

30

は1つの領域のみで動作して、ビットマップメモリ中の連続する領域を順に処理する。この際、現時点では対象外の領域に関するグラフィックス言語オブジェクトは一時無視して、ディスパッチャ330が当該領域の処理の開始を指示した時点になって処理を始める。このため、図9の構成では、一時的にグラフィックス言語コマンドを記憶するためのメモリのオーバーヘッドが必要となる。なお、ディスパッチャ330によって連続的に選択される領域を大きくすると、記憶すべきグラフィックス言語コマンド量が減少するため、このようなオーバーヘッドを少なくすることができる。

【0041】図10はグラフィックス言語プロセッサの別の構成を示す図であり、調整部440を設けて、マスク生成部421、背景生成部422、クリッパ425、ラストOPマシン426間のパイプライン処理の調整を行うものである。具体的には、図10の構成では、各モジュールはパイプライン化されたモジュールで、パイプラインに入力された情報順に処理を進める。各モジュールは、D.A. Patterson, J.L. Hennessy 著の「Computer Architecture: A Quantitative Approach」(1990, Morgan Kaufman 出版社)の第6章に一般的に示されているように、並列にかつ同時にパイプライン命令に基づいて処理を進める。すなわち、モジュールが同時に処理を開始して、処理の間の同期がとれるように調整部440が指示を出す。例えば、背景生成部422は背景をラストOPマシン426に直接出力することができるが、クリッパ425がマスク生成部421とラストOPマシン426との間に位置するため、マスク生成部421は対応するマスクオブジェクトをラストOPマシンに直接出力することはできない。そのため、調整部440は、クリッパ425からのクリッピングされたマスクと背景生成部422からの適切な背景との間の同期をとり、ラストOPマシン426でこれらの情報に対して処理を行うようにする。

【0042】ここで述べたグラフィックス言語プロセッサは、以下に示すような他の構成のプリンタと組み合わせて用いることもできる。以下に示されている方法を組み合わせて用いることで、ここで述べたグラフィックス言語コマンドをさらにDART表現に変換することが可能となる。DART表現を用いることでプリンタ資源を正確に表わすことができ、プリント処理に先立って、プリント処理に必要なプリンタ資源が足りるかどうかを判断することができる。資源が足りている場合にはプリント処理を開始し、プリンタ資源が足りない場合にはプリント処理が可能となるようにプリント処理を変更することができる。

【0043】<第2実施例>次に、前記第1実施例が好適に適用される一適用例として、前述のように画像をプリントする方法ならびに装置であって、画像をプリントする際に用いるプリンタ資源を予測することで、プリン

31

ト処理の開始に先立ってプリント処理を変更することのできるプリント方法ならびに装置を説明する。本実施例では、画像はフルスタ画像に変換される前には符号表現の形式で蓄えられており、この符号表現はフルスタ画像への変換時に用いる資源を正確に表わすことができる表現となっている。すなわち、画像の符号表現をフルスタ画像に変換するのに必要となるプリント資源を精度良く予測することができ、画像をプリントするのに十分な資源をプリンタが備えているかどうかをプリント処理に先立って判定することができる。この予測処理によって資源が足りないと判断されたときには、資源の再割り当てを行って画像のプリント処理を行う。

【0044】符号化された表現を用いることでプリント資源を動的にかつ適応的に利用することができるため、以下、このような表現を動的資源適応手法 (Dynamic Adaptive Resource Technology) 手法、略して「DART」表現と呼ぶことにする。例えば、解像度を操作する。例えば、図12に図式的に示す資源、CPUパワー、メモリ、バンド幅を考えてみる。ここで、CPUパワーとは、プリンタの制御部内のCPUがデータを操作し、1つの形式から他の形式に処理する速度と能力のことである。また、メモリとは、プリンタで利用可能なメモリ量のことである。バンド幅は、外部メモリ装置への通信リンクのバンド幅すなわちデータ転送速度のことである。なお、バンド幅には、アクセス時間、ディスクヘッド遅延、ディスク回転速度などの外部メモリ装置内での既知の遅延も含まれる。

【0045】いかなるプリンタでも、最大CPUパワー、最大メモリ、最大バンド幅で指定される点で定義される利用可能な資源の範囲がある。予測処理によって、DART表現のフルスタ画像への展開がこれらの資源範囲の1つでも越えると判断されると、資源の再割り当てが行われる。例えば、画像を記憶するメモリは足りないが、CPUパワーが余っていると判断されたとすると (図12の点A)、フルスタ画像へ展開する際に必要なメモリを少なく、必要なCPUパワーを多くするような別のDART表現へ変換することが可能である。すなわち、メモリをCPUパワーで置き換えることができる。このような再割り当てによって点Aは点A'に移動し、プリント資源の範囲内に収まるようになるため、プリント処理を実行することができる。

【0046】また、プリント解像度を減少させたり、中間調画像のグレイ階調数やカラー画像のカラー数を減少させることで、画質を変化させることも可能である。画質を変化させれば、CPUにさらなる負担をかけることなく、必要となるメモリ量を減らすことができる。このような再割り当てにより点Aは点A'に移動するため、プリント資源の範囲内に収まり、プリント処理を実行することができる。

【0047】逆に、予測処理によって、DART表現か

32

らフルスタ画像への展開に必要なCPUパワーが不十分で、ラスト画像を適切な時点でプリンタエンジンに転送することができない (すなわちデータアンダーフロー) と判断されることもある。この際、メモリが余っているとすると (図12の点Bで表わされる)、CPUパワーを多く必要とする画像部位を予めラスタリングするなどして、CPUパワーを利用可能なメモリで置き換えることができる。このような再割り当てによって点Bは点B'に移動し、プリント資源の範囲内に収まるようになるため、プリント処理を実行することができる。

【0048】同様に、CPUパワーならびにメモリとバンド幅とを置き換えることも可能である。例えば、DART表現からフルスタ画像への展開処理に最大のCPUパワーを用いているが、メモリ資源が足りないといった点Cのような場合には、画像の一部をオフラインで記憶しておき必要ときに通信リンクを介して取り出すなどの処理を行うことで、メモリをバンド幅に再割り当てすることができる。このような再割り当て処理によって、点Cは点C'に移動し、プリント資源の範囲内に収まるようになる。予測処理によって、ラスト画像データを適切な時点でプリンタエンジンに転送し、プリンタエンジンがデータアンダーフローを起こさないだけのバンド幅が十分にあることが判断されるのである。

【0049】図13は制御部11のブロック図である。なお、図13は制御部11中のすべての部位間の物理的接続を示すものではなく、制御部内での処理の流れを示す図である。図13に示されるように、制御部11はCPU41を備え、通信インタフェース12を介してページ記述言語などの画像データを受信する。すると、CPU41はGL変換部42でもって、ページ記述言語の画像情報をグラフィックス言語列に変換する。グラフィックス言語 (GL) とは、最もコンパクトなPDLレベルとフルスタ画像データレベルとの間に位置するような画像データ記述言語である。以下で詳細に説明するように、GL言語には画像オブジェクトなどのプリミティブと、このようなプリミティブをビットマップメモリに展開してフルスタ画像を生成する規則とが存在する。尚、グラフィックス言語 (GL) をグラフィックスストリーム言語 (GS) とも言う。

【0050】GL変換部42でのGL表現に基づいて、CPU41はDARTコード生成部43でDART表現を生成する。DARTコード生成部43で生成されたDART表現はラスト画像のコンパクトな符号化表現であり、DART表現からフルスタ画像への変換の際に用いる資源を精度良く予測することができる。また、DARTコード生成部43で生成されたDART表現は、GL表現のプリミティブの多くを用いている。しかし、ラスト画像全体に作用するGL表現とは異なり、DARTコード生成部43で生成されたDART表現は画像のバンド部のみに作用する。したがって、DART表現では

GL表現で用いるような複雑なプリミティブを利用しないことをも鑑みると、DART表現を用いることで資源利用を精度良く予測することが可能となる。なお、このような予測はGL表現では不可能である。

【0051】DARTコード生成部43において生成される各バンドごとのDART表現はメモリ44に蓄えられる。感光ドラム20上のレーザビーム16のスキャンに対応するデータをプリンタエンジン45が必要とするときに、メモリ44に蓄えられている表現はレンダ一部46によってフルスタバンド画像に展開され、転送部47によってプリンタエンジンに転送される。

【0052】図14は、DARTコード生成部43において生成されるDART表現に応じて、画像をバンドに分割する様子を示す図である。図14に示されるように、ページ画像はプリンタエンジンのスキャン方向（ここでは水平方向）に対応するバンドに分割される。一般に、 $8 \cdot 1/2 \times 11$ インチの画像を効率良く処理するためには27個程度のバンドに分割すれば十分である。なお、図14では説明の簡潔さを考慮してすべてのバンドを図示していない。

【0053】メモリ44は、従来のプリンタのビットマップメモリに対応するメモリ領域である。ここで、メモリは待ち行列（キュー）として利用される。すなわち、DART表現が生成されるとメモリを割り当て、レンダリングされてプリンタエンジンに転送されるとメモリを解放する。新たなページが生成されると、空きメモリの最初の位置にメモリが割り当てられる。ここで、物理的なメモリの最下部と最上部とは連続するようにメモリは割り当てられる。ページに対してすべてのDARTオブジェクトとアプリケーションとが生成され、バンドのラスタ記憶がメモリに付加されると、現在のページメモリの後部に次のページが割り当てられる。ページの転送がすべて終了すると、バンドラスタ、オブジェクト、ページ情報構造などを解放し、再利用可能な状態に戻す。

【0054】図14に示される各バンドはDARTコード生成部43において逐次的に処理される。すなわち、例えばDARTコード生成部43では最初にページiのバンド0を処理し、得られたDART表現をメモリ44の最上部に蓄積する。次いで、DARTコード生成部43はページiのバンド1を処理し、得られたDART表現をメモリ44の続き領域に蓄積する。このような処理をページiの最終バンドKまで続けると、全ページのDART表現がメモリ44に蓄積される。この処理では、ページiのフルスタ画像はまだ存在せず、バンドDART表現が得られるのみである。

【0055】DARTコード生成部43において各バンドが処理されると、CPU41は予測機能を用いて、プリンタ10の資源の範囲内でメモリ44に蓄えられているDART表現をフルスタ画像に展開できることを確認する。このような予測が可能となるのは、DART表

現におけるプリミティブ構造と、DART表現をフルスタ画像に展開する符号の信頼性による。プリンタ資源の範囲内で各バンドの展開が可能であるとすると、CPU41はメモリ44のページiの最後に2つの空白バンドラスタ（すなわち、バンドラスタ0とバンドラスタ1）を付加する。ここで、各空白バンドラスタのサイズは、所望のプリント解像度での1バンドのフルビットマップ画像のサイズである。

【0056】続いて、DARTコード生成部43は次のページすなわちページi+1のバンドDART表現を生成する処理に進む。これらの処理の間、レンダ一部46は空白バンド上にページiのバンドDART表現をフルスタ画像に展開する処理を開始する。まず、レンダ一部46はバンド0をバンドラスタ0に展開する。続いて、レンダ一部46はバンドラスタ1にバンド1のDART表現をフルスタ画像として展開する。バンド1が展開されている間、転送部47はバンドラスタ0中のフルビットマップラスタ画像をプリントエンジン45に転送し、バンドを出力する。バンドラスタ0のエンジン45への転送が終ると、レンダ一部46はバンドラスタ1中のフルスタ画像を展開してバンド1を出力する。ここで、十分な資源が利用可能であり、プリントエンジンでデータのアンダーフローが起こらないように所定の時間内でフルスタ画像が生成できることは、プリント処理に先立って予測機能を用いて確認されている。

【0057】続いて、転送部47はバンドラスタ1中のフルスタ画像を転送する。この間、レンダ一部46はバンド2中のラスタ表現をフルビットマップラスタ画像に展開する。これらの処理をページiのすべてのバンドがレンダ一部46によって展開され、転送部47によってプリントエンジン45に転送されるまで繰り返す。ページiのDART表現がフルスタ画像に展開され、転送されて出力される間、空きのCPU時間を用いて次ページすなわちページi+1のPDLを上記のページiと同様の手順でDART表現に変換する処理が行われる。

【0058】上述の処理において、CPU41が予測機能をもって、DART表現からフルスタ画像への展開が利用可能な資源の範囲を越えていると判断した場合には、資源の取捨選択あるいは再割り当てが行われる。例えば、DART表現がプリント情報のフルスタ画像以上である場合、すなわちDART表現がフルスタ画像と比べて非効率になっている場合で、CPUパワーが利用可能である場合には、DARTコード生成部43において別のよりコンパクトなDART表現が生成される。逆に、CPU41がDART表現からフルスタ画像への展開処理をバンド情報処理に要する時間内に行うことができるだけのCPUパワーを有していず、プリントエンジンにおいてデータのアンダーフローが生じる場合には、障害となるバンド（障害となるバンドを含むすべてのページ）を前もってラスタリングし

35

ておき、ラスティングされたデータが適切なタイミングで転送部47に転送される。

【0059】図15は全体の処理の流れを示す図である。図15中の処理ステップは制御部11中の不図示のメモリに蓄えられており、CPU41で実行される。ステップS601では、制御部11は通信ポート12から画像データを入力する。画像データは一般にホストコンピュータによって生成され、ホストコンピュータから制御部11へはページ記述言語などのコンパクトな符号で転送される。ステップS602では、GL変換部42がPDL画像データをGL画像データに変換する。ステップS603では、DARTコード生成部43においてGLデータからDART表現が生成され、メモリ44に蓄えられる。DART表現が生成されると、ステップS604でバンドごとに分割され、ステップS605でDART表現をフルスタ画像に展開するのに必要な資源を予測する。すなわち、各DART表現ごとに、予測機能でもって(a) DART表現を展開するのに必要なCPUパワーと(b) DART表現を蓄えるのに必要なメモリとを予測する。プリンタ10がバンド幅が制限された通信線を介してCPU41に接続される外部メモリを備えているときには、各DART表現ごとのバンド幅の要求に関しての第3の予測機能が用いられる。

【0060】ここで述べた資源、すなわちCPUパワー、メモリ、バンド幅とは、プリンタ10が備える資源の中で代表的なものである。これ以外の資源に対しても、対応する予測機能を用いて、CPU41で資源の利用を予測することができる。ステップS606では、予測結果に基づいて、現在のバンドのDART表現をバンドラスタのフルスタ画像に展開する際に必要となる資源が得られるかどうかを、CPU41が判断する。なお、このステップS606の処理はDART表現が生成されるたびに実行することも、あるいはすべてのバンドのDART表現が生成された時点で実行することもできる。DART表現に対してメモリ資源が足りないかどうかを判断する場合には、ステップS606の処理を周期的に、例えば新たなDART表現オブジェクトが生成されるごとに実行することが好ましい。ステップS606において現在のバンドを展開するのに必要な資源が十分に得られるとCPU41が判断した場合には、処理をステップS607に進め、現在のページに他のGL符号があるかどうかをCPU41が調べる。ページに他のGL符号が存在する場合には、処理をステップS603に戻し、残りの画像データの処理を実行する。

【0061】ページ中のすべてのバンドに対するDART表現が生成されると、ステップS607からステップS608に処理を進め、2つの空白バンドラスタをメモリ44に蓄えられているDART表現に付加する。処理がこのステップに進んだ場合には、プリンタの資源が、DART表現をバンドごとにフルスタ画像に展開する

36

のに十分であることが確認されており、プリンタ資源の範囲を越える危険性はない。すなわち、CPUパワーが十分であり、プリンタエンジンにおいてデータのアンダーフローは生じない。

【0062】ステップS610では、図14において説明したように、バンドごとにDART表現を2つのバンドラスタのうちの1つに交互に展開する。次いで、ステップS611で、転送部47がラスタ画像をプリンタエンジン45に転送する。ステップS610とS611とは、図14において説明したように、同時にそして逐次的に実行される。すなわち、転送部47において1つのバンドラスタ画像を1つのバンドラスタから転送し終えた時点で、レンダ一部46では他のバンドのDART表現を他のバンドラスタに展開する処理が終了する。

【0063】なお、2個以上の空白バンドラスタを用いることも可能で、この際にはレンダ一部46における展開処理は転送部47の固定数分のバンドの転送以前に終了する。現在のページが出力されると、あるいはステップS610とS611の展開処理と転送処理中にCPU時間がとれると、図15の全体の処理の流れにしたがってCPU41は次のページの処理に進む(ステップS612)。

【0064】ステップS606において、プリンタ資源が不十分であるとCPU41が判断した場合には、現在のDART表現を展開して蓄積するのに必要な資源がプリンタが持つ資源以上であることがわかる。そこで処理をステップS613に進め、CPU41が資源の再割り当てを行い、プリンタ資源の範囲内に収まるようにDART表現を修正する。まず、障害となるバンドに対して、別のDART表現を選択する。例えば、メモリ資源が足りなければ、多くのメモリを必要とするDART表現を比較的少ないメモリで良い別の表現に変更する。同様に、CPU資源が足りない場合には、CPUを多く必要とするDART表現を比較的少ないCPUで実行可能な別の表現に変更する。例えば、障害となるバンドに対して前もってラスティングしてメモリ44に記憶することで、さらなるCPU処理を行うことなくプリンタエンジンに出力することができる。

【0065】ステップS614では、別のDART表現を用いたときに必要となる資源を再予測する。ここで、必要となる資源が資源の範囲内で収まる場合には、処理をステップS607に進め、上述の処理を実行する。一方、必要となる資源がプリンタ資源の範囲以上であるときには、必要となる資源をプリンタ資源の範囲内に収める処理をさらに行う(ステップS613とS614)。

【0066】これに対して、DART表現を資源の範囲内に収めることができない場合には、出力に先立ち画像を変化させる処理を行う。画像を変化させることで、自動的に必要となる資源を減らすことができる。ここでの実施例では、解像度(あるいは画素密度)を600dp

37

iから300dpiに減少させることで画像を変化させる。画像を変化させる方法としては、この他にもいくつかの手法が考えられる。例えば、中間調画像の場合には、グレイ階調の数を減少させることで画像を変化させることができる。また、カラー画像の場合には、カラーの数を減らして画像の変化を図ることができる。

【0067】ステップS616では、解像度を減少させる準備として、メモリ44中のすべてのページに対する処理を終了させる。すなわち、これまでのページのDART表現をステップS610とS611のように展開して転送する。ステップS617では、DARTコード生成部43ですでに生成されているDART表現を展開して、画像の解像度を減少させる。DART表現はメモリ44のものと位置に低解像度で展開される。また、処理されていないGLはメモリ44で変換ラスタ画像に変換される。このような変換処理は、変換ラスタ画像への直接的変換、もしくはDART表現を生成してから展開するという間接的変換のどちらかで行われる。ステップS618では、転送部47において変化したラスタ画像をプリンタエンジンに転送する。この際、画像の変化処理が必要な場合には、変化の影響を最小にするために何らかの画像強調処理を施すことが好ましい。例えば、解像度を減少させる場合には、変化した画像を転送する際に（米国特許番号4,878,068に示されるような）平滑化回路を用いて、出力された画像の画質向上を図ることが好ましい。ステップS619では、次のページに対して同様の処理（変化処理は除く）を行う。

【0068】ステップS613からS619までの処理は、DART表現に十分な資源が得られるように資源の再割り当てを繰り返し試みる処理である。しかし、資源再割り当ての最も簡潔な方法として他の好適な実施例としては、資源が足りないと判断された時点で直接画像を変化させる手法がある。すなわち、カラー、グレイ階調、画素密度（すなわち解像度）を変化させる手法である。例えば、600dpiラスタ画像に展開される白黒画像の場合には、解像度を300dpiまで減少させると、メモリ44にフルラスタ画像を蓄積することができる。この場合、DART表現は画像データの変化したフルラスタ画像に変換され、フルラスタ画像が転送部47を介してプリンタエンジンに転送される。この際にも、（米国特許番号4,878,068に示されるような）画像平滑化手法を用いて、低解像度画像の画質を改善することもできる。

【0069】図16は資源が足りないと判断された時点で直接画像を変化させる処理の流れを示す図である。図16において、ステップS601からS604は図15と同一の処理であり、ここでの説明は省略する。ステップS605では、DART表現をフルラスタ画像に展開するのに必要な資源を予測する。図15の実施例と同様に、DART表現ごとに、CPUパワー、メモリ使用

38

量、バンド幅などの予測を行う予測機能を備える。また、変換画像をビットマップメモリにラスタリングするのに十分なメモリがあることを確かめるために（変化が必要な場合には）、フルビットマップ画像を低解像度でラスタリングするのに必要なメモリ使用量を予測する。

【0070】ステップS607からS608は図15と同一の処理であり、ここでの説明は省略する。ステップS606において、必要となる資源が足りないと判断された場合には、あるいはフルビットマップ画像を低解像度でラスタリングするのに必要なメモリ量が不足すると判断された場合には、画像を直接変化させて、必要となる資源をプリンタ資源の範囲内に収める。この画像の変化処理は、ステップS620、S621、S622、S623で行われる。なお、これらはそれぞれ図15のステップS616、S617、S618、S619と同一の処理である。

【0071】＜DART表現の予測機能＞以下、種々の好適な画像のDART表現ならびに各表現に対応する予測機能について説明を加える。表現で用いるプリミティブはオブジェクトと背景との2つである。DART表現は、これらの様々なオブジェクトの様々な背景への一連のアプリケーションから構成される。背景はビットマップタイルや画像から成る。ビットマップタイルは変換部42からのGL画像表現から得られる。カラーのビットマップタイルへのマッピングは、カラーをそれぞれのタイルのカラー面にマッピングすることで行われる。一方、画像は画像表現であり、展開と予測処理とを高速に行うために最適化されている。ラスタ表現で用いられるオブジェクトプリミティブは、これのフルラスタ画像への展開が信頼性高くまた精度良く予測できるという点で好ましい。オブジェクトプリミティブの好ましい組として、ビットマップ、ランレングスシーケンス、台形シーケンス、ボックス、高速境界符号ビットマップが挙げられる。これらのオブジェクトの各々に対して、以下で詳細に定義する。

【0072】図17は、DART符号の画像表現をフルラスタビットマップに展開するアプリケーションを示す図である。図17では、タイルあるいは画像で定義される背景がアプリケーションマシーンに入力される。また、ビットマップ、ランレングス、台形、ボックス、高速境界符号マップなどのオブジェクトもアプリケーションマシーンに入力される。アプリケーションマシーンは、対象バンドのビットマップ中の特定の領域で背景上にオブジェクトを適用し、ビットマップの初期ラスタを図17に示す処理後のラスタに変換する。このアプリケーションマシーンは、第1実施例のラスタROPマシーンに相当する。

【0073】それぞれのDARTオブジェクトごとに異なる展開手法が用いられ、各展開手法はDARTオブジェクトをできるだけ効率良く展開するように構成されて

いる。展開手法は制御部のCPUの速度と符号のコンパクトさに応じて最適化されていることが好ましい。適切な展開手法は、Addison-Wesley社から1987年に発行されたForley, van Dam, Feiner and Hughesの「Computer Graphics: Principles and Practice」(第2版)に示されている。

【0074】DART表現の予測部は、DART表現を*

$$\text{MemoryRequired}_i = \sum_{k=1}^n \text{BandSize}_k + 2 \times \text{BandRasterSize}$$

【0076】で表わされる。ここで、 BandSize_k はk番目のバンドを展開し終えたときに解放されるメモリ量、 BandRasterSize はページメモリの最後に付加されるフルラスタされたバンドラスタサイズ、nは現在記憶しているDARTオブジェクトのバンド数である。CPU41はこのメモリ量が利用可能なメモリ量を越えないことを確認する。また、高解像度で出力するのに必要な資源が*

$$\text{DegradedMemRequired}_i = \sum_{k=1}^n \text{BandSize}_k + (i+1) \times$$

$\text{BandRasterSizeDegraded}$

【0078】を求める。ここで、 $\text{BandRasterSizeDegraded}$ は低解像度でのバンドラスタサイズである。DART表現を展開するのに必要な時間(CPUパワー)の予測に関しては、展開処理がオブジェクトと背景の2つを生成する処理、ならびにオブジェクトと背景とのラスタへのアプリケーションとを含むため、展開に要する時間はこれらの処理時間の和として求められる。そこで、予測部ではこれらの点を考慮して予測を行う。実際には、展★

$$\text{Time} = (\text{ApplicationTime} + \text{BackgroundTime}) \times$$

OverheadFactor

【0081】のように表わされる。ここで、 ApplicationTime はDARTオブジェクトの適用に要する時間、 BackgroundTime は背景を展開するのに要する時間、 OverheadFactor はシステムオーバーヘッド処理を考慮した上述のオーバーヘッド係数である。ここで、 BackgroundTime は展開される背景領域に関係し、また背景の種別と展開手法にも関係する値である。

【0082】ビットマップオブジェクトは、その名の通り、単なるオブジェクトのビットマップラスタ画像である。したがって、図18に示されるように、オブジェクト71は、IDラベル(ID LABEL)とオブジェクト71のビットマップラスタ画像とから成る。ビットマップオブジェクトの部位72のバンドへのアプリケーションも、☆

$$\text{TimeBIMap} = (N_1 \times \text{ht} + \text{Background}) \times \text{OverheadFactor}$$

【0085】で行われる。ここで、 N_1 はビットマップオブジェクトの各列を展開するのに必要なCPUサイクル数に関連する実行係数である。図19はランレングスオブジェクトを示した図である。ランレングスオブジェクトはセグメントに分割して記述され、各セグメントは◆50

*記憶するのに必要なメモリと、DART表現を展開するのに必要な時間(CPUパワー)とを予測する。一般に、画像中のi番目のバンドを展開しているときのDART表現を記憶するのに必要なメモリ量は、

【0075】

【数1】

10※足りない場合に画像を低解像度で展開するといった図16のような実施例では、CPU41は必要なメモリ量が低解像度で画像を展開するのに必要なメモリ量以下であることを確認する。すなわち、

【0077】

【数2】

★開処理におけるシステムオーバーヘッド処理を考慮してオーバーヘッド係数をさらに加える。

【0079】以下、オブジェクトとそれに関連した予測処理について詳細に説明する。一般に、CPUパワーの予測式は、

【0080】

【数3】

☆図18に示されている。図示のように、オブジェクトの開始点から距離dispの位置から始まるバンド上にオブジェクトが適用され、バンド内の位置by, xにhtの高さで展開される。73がアプリケーションであり、オブジェクトが適用される背景を定義するモード定義MODE、オブジェクトが適用されるバンド内の座標x, by, オブジェクトが適用される高さhtとが含まれる。さらに、オブジェクトのIDラベルと、オブジェクト内の距離DISPも指定されている。

【0083】ビットマップアプリケーションを展開するのに必要なCPUパワーの予測式は、

【0084】

【数4】

◆連続するスキャンラインを記述し、各スキャンラインは1つのランレングスから構成される。すなわち、図19に示されるように、ランレングスオブジェクト81はセグメント1からセグメント4までを含む。

【0086】82はランレングスオブジェクトのDAR

41

T表現を示したものである。オブジェクトには、IDラベル(ID LABEL)と、セグメントの展開時間の予測に関する情報を含むセグメント番号とが含まれる。この情報は予測処理において用いられる。なお、各セグメントは所定数までのランレングスを記述したものであり、各セグメントはバイト単位に配置される。

【0087】セグメントでは連続するランレングスをセグメント状態を順々に変えた異なる形で定義する。ここで、セグメント状態は2つの境界状態(左と右)の対から構成される。また、各輪郭は位置値 x と更新値 dx から構成される。セグメントの開始位置では、 $x_{l0} = dx_{l0} = x_{r0} = dx_{r0} = 0$ である。

【0088】各スキャンライン i ごとにセグメントは更新ステップ($step_i = \langle advl_i, advr_i \rangle$)を定義して、以下のように更新する。

【0089】

【数5】

$$dxl_i = dxl_{i-1} + advl_i$$

$$dxr_i = dxr_{i-1} + advr_i$$

$$xl_i = xl_{i-1} + dxl_i$$

$$xr_i = xr_{i-1} + dxr_i$$

$$TimeRunLength = (N_2 \times ht + N_3 \times area + Background) \times$$

Overheadfactor

【0093】で行われる。ここで、パラメータ N_2 、 N_3 はビットマップオブジェクトと同様のパラメータである。図20は台形オブジェクトを示す図である。台形オブジェクトは、ランレングスオブジェクトとは異なり分割されていない。これは、台形シーケンスを通常繰り返して用いることはないと考えられるためである(台形オブジェクトは大きなラスタ領域を記述するオブジェクトである)。

【0094】各オブジェクトは連続するスキャンラインを記述し、各スキャンラインは1つのランレングスから構成される。台形シーケンスでは連続するラン長をDDAアルゴリズムで記述する。状態は2つの境界状態(左と右)の対とから構成される。また、各輪郭状態は位置 x と傾きとから構成される。

【0095】台形シーケンスの開始位置は、 $xl = slope\ l = xr = slope\ r = 0$ である。オブジェクトはアクションのシーケンスとして定義される。ここでのアクションは「修正」と「ペイント」とのどちらかである。「修正」アクションはベア $modify = \langle modx, modslope \rangle$

として示され、以下のように状態を更新する。

【0096】

【数7】

42

*【0090】図19は、ランレングスオブジェクト81の部位のバンド84へのアプリケーションも示している。図示のように、バンド84内の位置 by 、 x にランレングスオブジェクトの部位82が高さ ht まで適用される。このアプリケーションを符号化した表現が83に示されているが、ランレングスオブジェクトが展開される背景の種類を定義するモード(MODE)が含まれている。さらに、 X 、 by 、 ht の座標情報、IDラベル、セグメントインデックスからのセグメント(DESEG)、セグメントの開始位置からのオフセット(DOFF)が含まれる。

【0091】ランレングスオブジェクトを展開するのに必要なCPUパワーの予測式は、

【0092】

【数6】

20

*

※

 $x += modx;$ $slope += modslope;$

30

【0097】「ペイント」アクションはランレングスの繰り返し数を定義するもので、以下のステップを定義された回数繰り返す。

【0098】

【数8】

 $xl += slope_l;$ $xr += slope_r;$ $y ++;$ $IssueRunLength(xl, xr, y);$

40

【0099】繰り返し数がゼロの「ペイント」アクションは台形シーケンスの終わりを示す。図20は、台形オブジェクトの部位92のバンド94へのアプリケーションをも示している。図示されているように、部位92はバンド中の位置 by 、 x に高さ ht まで適用される。アプリケーションは93に示されているが、オブジェクトが適用展開される背景を定義するモード(MODE)、バンド中の x 、 by 、 ht 座標情報が含まれる。また、オブジェクトのIDラベル(ID LABEL)も含まれる。台形オブジ

※50

43

44

ェクトは分割されたオブジェクトではなく、台形全体が距離 h_t に展開されるため、これ以外の情報は必要ない。

【0100】台形アプリケーションを展開するのに必要*

$$Time_{Trapezoid} = (N_4 \times h_t + N_5 \times area + Background) \times$$

OverheadFactor

【0102】で行われる。ここで、 N_4 、 N_5 ビットマップオブジェクトと同様のパラメータである。図21はボックスオブジェクトを示す図である。長方形ボックスは多く見られるオブジェクトであり、最適化された表現となっている。ここでは、図示の通りアプリケーション自身がボックスを定義するため、それ自体のオブジェクトは必要ない。

【0103】ボックスオブジェクトのバンド124へのアプリケーションも図21に示されている。図示の通り、ボックスオブジェクトは位置 b_y 、 x にバンド中の※

$$Time_{Box} = (N_6 \times h_t \times w + Background) \times OverheadFactor$$

【0106】で行われる。ここで、 N_6 はビットマップオブジェクトと同様のパラメータである。図22は、高速境界符号(FBE)オブジェクトを示す図である。FBEセグメントは連続するスキャンラインを定義するもので、各スキャンラインは複数のランレングスから構成される。

【0107】ここでも、セグメントインデックスは展開時間の予測に関する情報(dt-fbePred-t記録)を含み、予測処理で用いられる。なお、ここでの情報はランレングスオブジェクトのものに比べて大きいものとなる。セグメントでは連続するランレングスをセグメント状態を順々に変えた異なる形で定義する。

【0108】ここで、セグメント状態は複数の境界状態から構成される。また、各輪郭は位置値 x と更新値 dx から構成される。セグメントの開始位置では、 $x[k] = dx[k] = 0$ (k の全ての値で)である。

【0109】また、セグメントはコマンド列であり、以下の中から選択される。

adv: RLDeqsの場合に当該輪郭を進める。

close: 次の2つの輪郭を削除する。

open: 現在の輪郭と次の輪郭との間に2つの新しい輪郭を挿入する。

各スキャンライン i ごとにセグメントは更新ステップ

$step\ i = \langle line\ change, comm_1, comm_2, \dots, comm_n \rangle$

を定義し、セグメント状態を変換するのに用いる。ここで、Linechangeは前のスキャンラインに対する輪郭ペア数の変更(増加あるいは減少)、あるいはセグメント境界のリセットを指定するものである。また、 $comm_1, comm_2, \dots, comm_n$ はlinechangeに応じて輪郭数を変更するコマンドである。linechangeで輪郭ペア数が負になると、セグメントの終わりと判断される。

*なCPUパワーの予測は、

【0101】

【数9】

※高さ h_t 、幅 w まで適用される。アプリケーションは123に示されているが、ボックスが適用される背景を定義するモード(MODE)や、展開座標 x 、 b_y 、 h_t を含む。空白ワードの後に、ボックスの幅(WIDTH)が指定される。

【0104】ボックスオブジェクトを展開するのに必要なCPUパワーの予測式は、

【0105】

【数10】

★【0110】図22は、FBEオブジェクト131の部位132のバンド134へのアプリケーションも示している。図22に示されるように、位置 b_y 、 x のバンド内に部位132が、セグメント3の開始位置からのオフセット $doff$ から高さ h_t まで適用される。アプリケーションは116に示されているが、FBEオブジェクトが適用される背景の種類を定義するモード(MODE)が含まれている。さらに、 X 、 b_y 、 h_t の展開座標、オブジェクトのIDラベル(ID LABEL)、展開されるセグメント(dseg)、セグメントの開始位置からのオフセット(dof)が含まれる。

【0111】FBEオブジェクトを展開するのに必要なCPUパワーは、各セグメントに必要なCPUパワーの和として予測される。ここで、各セグメントごとのCPUパワーは展開されるラン長の数とセグメントの高さに基づいて決定される。

【0112】

【発明の効果】本発明により、プリンタ内の資源をより効率的に利用することで、利用可能な物理的な資源を増加させることなくプリンタの性能を向上させることができる。また、装置に依存しないPDLコマンドを対応するラスタ画像に変換するPDLインタプリタをプリンタごとに個別に用意する必要をなくし、汎用性を高めることができる。

【0113】[付録A]

グラフィックス言語コマンド命令セット

本付録はGLコマンド命令セットをまとめたものであり、システムへの作用や動作意味などの観点から命令を定義したものである。以下の表記はグラフィックスプロセッサが実行するスタック操作を示している。

【0114】命令 $arg_1, arg_2, \dots, arg_n$ 命令 $result_1, \dots, result_k$

45

ここで、列 $arg_1, arg_2, \dots, arg_n$ は命令が用いるスタック (arg_n が最上位) 上の変数リストである。命令を実行すると、変数リストは結果の列 $result_1, \dots, result_k$ に置き換えられる。

<算術ならびに関係演算>システムスタック上で算術演算や関係演算を実行する際に、これらの命令を用いる。

【0115】add number1 number2 gsop.add number (= number1 + number2) …加算

sub number1 number2 gsop.sub number (= number1 - number2) …減算

mul number1 number2 gsop.mul number (= number1 * number2) …乗算

div number1 number2 gsop.div number (= number1 / number2) …除算 (整数)

mod number1 number2 gsop.mod number (= number1 mod number2) …法演算

neg number gsop.neg number (= -number) …負演算

shright number1 number2 gsop.shright number (= number1 >> number2) …右シフト

shleft number1 number2 gsop.shleft number (= number1 << number2) …左シフト

<他の算術/関係命令>GLコマンドは、ベクトル計算 (固定座標) や必要となる三角関数 (高速近似) など、複数の拡張演算を備えている。

【0116】<スタック操作>これらの演算はシステムスタックを操作するためのものである。

loadq_o gsop.loadq_0 num(=0)

loadq_1gsop.loadq_1 num(=1)

loadq_2gsop.loadq_2 num(=2)

loadq_3gsop.loadq_3 num(=3)

loadq_4gsop.loadq_4 num(=4)

loadq_5gsop.loadq_5 num(=5)

loadq_6gsop.loadq_6 num(=6)

loadq_7gsop.loadq_7 num(=7)

システムスタックを高速にロードして最も頻繁に用いる算術値を取り出すためには、高速ロードファミリ命令を用いる。

【0117】pop anyvalue gsop.pop …スタックのトップの削除

dup anyvalue gsop.dup anyvalue anyvalue …スタックのトップの複製

exch anyvalue1 anyvalue2 gsop.exch anyvalue2 anyvalue1 …スタックのトップ2つのセルの交換

<他のスタック操作命令>GLコマンドは、インデックスやコピーなどのスタック操作のための拡張演算を複数備えている。

【0118】<記録>永続的な情報を記録するために記録命令を用いる。記録中の情報は配列として構造化されている。グラフィックス言語コマンドはGLオブジェクト記憶中のGLオブジェクトとして記録する (以下参

46

照)。現記録という概念を設けることで、GLコードを圧縮して記録することができる。すなわち、オブジェクトを現記録として宣言すると、すべてのロードや記録演算はそのオブジェクトに対して行われる。

【0119】各フレームのはじめに、システムはデフォルト記録を有する。

record object|d| gsop.record …オブジェクトobject|d| を現記録とする。

load displ gsop.load …記録位置displ のメモリ内容をスタックにプッシュする

store value displ gsop.store …スタックのトップのvalue を記録位置displ にポップする

<配列操作>これらの命令は配列を生成し、スタック上の配列要素にアクセスするものである。

【0120】<オブジェクト操作>これらの命令はGLオブジェクトを生成し、操作するものである。

objdelete objhandle gsop.objdelete …メモリからオブジェクトhandleを削除する

objmke objhandle size gsop.objcopy …オブジェクトhandleに対して「削除」処理を実行し、サイズsizeの新たなオブジェクトを生成する

markarray gsop.markarray + n + b1 + ... + bn inlinehandle

markbitmap gsop.markbitmap + n + b1 + ... + bn inlinehandle

marktrap gsop.marktrap + n + b1 + ... + bn inlinehandle

markproc gsop.markproc + n + b1 + ... + bn inlinehandle

これらの命令はinlineオブジェクトを宣言し、今後の参照を鑑みてスタックトップへのinlinehandle操作を残す。

【0121】objdef inlinehandle objhandle gsop.objdef …オブジェクトobjhandle に対して「削除」処理を実行し、指定サイズの新たなオブジェクトを生成し、inlinehandleの内容を新たなオブジェクトにコピーする。

<制御>これらの命令はGLプロセッサの実行処理の流れを制御するものである。

【0122】call executableArray gsop.call …操作executableArray によって示される処理を呼び出す。return gsop.return …現在処理中の処理から実行処理の流れを戻す。GLプロセッサが処理実行中でない場合には、この命令により現フレームの処理が終了する。

【0123】if bool execArraytrue gsop.if

if else bool execArraytrue

execArrayfalse gsop.iffalse

…これらの演算は、値bool (ゼロが偽、非ゼロが真) に基づいて実行する処理を適宜変更する。

【0124】loop number execArray gsop.loop …exe

47

cArray をnumber回繰り返して実行する。負の繰り返し回数は無限ループを示す。

break gsop.break

continue gsop.continue

…現在実行中のループのブレイクあるいは続行である。

【0125】switch number gsop.switch…ケース値numberに基づいてそれぞれ別のプログラム部位を実行する。ケース値はB = {0~255}に制限されている。

<内容操作>

context contextNumber gsop.context…現在の内容をcontextNumber にセットする。可能な内容の数は固定されている。

【0126】contextDef executableArray contextNumber gsop.contextDef…処理executableArray ごとに内容contextNumber を定義する。内容処理はswitch文を含み、コードをアクションにマップするために用いられる。

<処理定義>処理は単なる実行可能な配列である。処理操作は配列操作あるいはオブジェクト操作である。

<システムサービスアクセス>システムサービスは、システム処理操作が特別なセットである場合以外は、通常の処理のようにアクセスされる。

【0127】<マスク生成>これらの命令はGLからのプリミティブマスク生成部を起動する。すべての生成部を起動する前に、GLプログラムは生成部パラメータに対応する状態情報をセットしなければならない。

PixelBit bitmapHandle x y gsop.PixelBit…画素ビット演算bitmapHandleを位置<x, y>で実行する。

【0128】Trapezoid trapezoidHandle gsop.Trapezoid…台形マスク生成部を起動する。

RLE gsop.RLE…ランレングス符号マスク生成部を起動する。

<ライン生成マスク>これらの命令はライン展開のためにマスク生成部を起動する。

【0129】これらのマスク生成部はGL状態変数を用いて、位置、ライン幅などを間接的に定義する。

<入力列へのアクセス>この種類のプリミティブ要素は入力列にアクセスし、スタックあるいは配列スタックにオブジェクトを生成する。

【0130】getc1 gsop.getc1 num (= b1)

getc2 gsop.getc2 num (= (b1 << 8) + b2)

getc3 gsop.getc3 num (= (((b1 << 8) + b2) << 8) + b3)

getc4 gsop.getc4 num (= ((((b1 << 8) + b2) << 8) + b3) << 8) + b4)

…入力列から次の1から4バイト(b1, b2, b3, b4)を取り込み、スカラー整数値としてスタック上にプッシュする。

【0131】<背景生成>

image a b c d e f image gsop.Image…背景画像配列

48

として画像をセットする。変換係数[a b c d e f]はラスタ画素の画像配列要素へのマッピングを示す。

image 0 gsop.Image…背景画像配列を空にリセットする。

【0132】gray grayLevel gsop.gray…背景をレベルgrayLevel のグレイにセットする。

betile bitmapHandle gsop.bgtile…bitmapHandleをもってタイルすべき背景をセットする。

bgmodify offsetX offsetY rotation gsop.bgtile…背景タイルを<offsetX, offsetY>だけシフトさせて、rotationの角度分回転させて修正する。

【0133】<状態演算>

screen objHandle gsop.screen…objHandle のディザ画面をセットする。

clip trapHandle gsop.clip…trapHandleをクリッピング領域に付加する。

【0134】clip 0 gsop.clip…クリッピング領域をゼロにする。

<アグマティクス>GLインタプリタはこの種類の命令を無視する。この語用命令は他のサブシステムやツール(デバッグツール、生成ライブラリなど)において用いられるものである。

【0135】

pragma gsop.pragma + {n} + b1 + b2 ... + bn

GLインタプリタはこの命令を構成する(n+2)バイトを無視する。命令pragmaはコメントとして、デバッグ目的の同期タグとして、グルーパタグなどとして用いる。

【図面の簡単な説明】

【図1】ビットマップ・ラスタメモリへの画素のラスタライズ処理を示す図である。

【図2】PDLプリンタの処理を説明する図である。

【図3】本実施例のプリンタの構成を示す図である。

【図4】本実施例のグラフィックス言語プロセッサを示す図である。

【図5】本実施例におけるページ記述言語とグラフィックス言語との間のインタフェースを説明する図である。

【図6】マスクがラスタ上の背景に適用される様子を説明する図である。

【図7】本実施例のグラフィックス言語プロセッサにおけるマスク生成部と背景生成部とを詳細に示す図である。

【図8】逐次型のグラフィックス言語プロセッサの構成を示す図である。

【図9】バンド分割型のプロセッサの構成を示す図である。

【図10】パイプライン型のプロセッサの構成を示す図である。

【図11】プリントサイクル中のCPUとメモリの利用

率を示す図である。

【図12】プリントが利用可能な資源の範囲を示すグラフ図である。

【図13】図3のプリンタの制御部の構成例を示すブロック図である。

【図14】図3のプリンタにおけるメモリ構成例を説明する図である。

【図15】図3のプリンタの処理手順例の流れを示すフローチャートである。

【図16】図3のプリンタの処理手順例の流れを示すフローチャートである。

【図17】本実施例の符号化画像表現例を説明する図で

ある。

【図18】図17に示されるオブジェクト画像例を説明する図である。

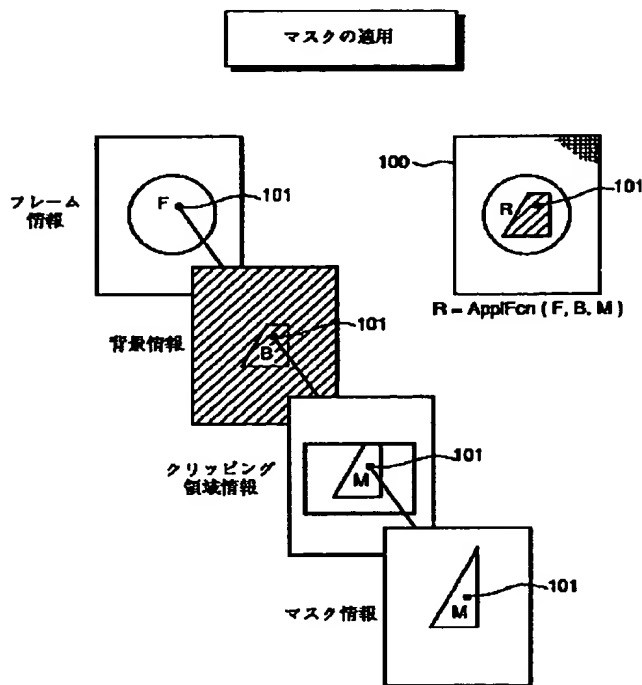
【図19】図17に示されるオブジェクト画像例を説明する図である。

【図20】図17に示されるオブジェクト画像例を説明する図である。

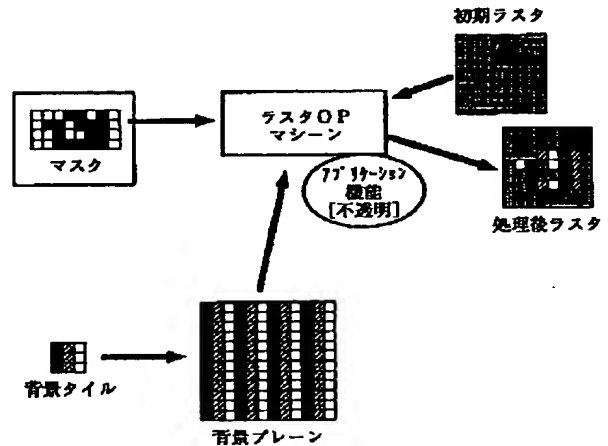
【図21】図17に示されるオブジェクト画像例を説明する図である。

【図22】図17に示されるオブジェクト画像例を説明する図である。

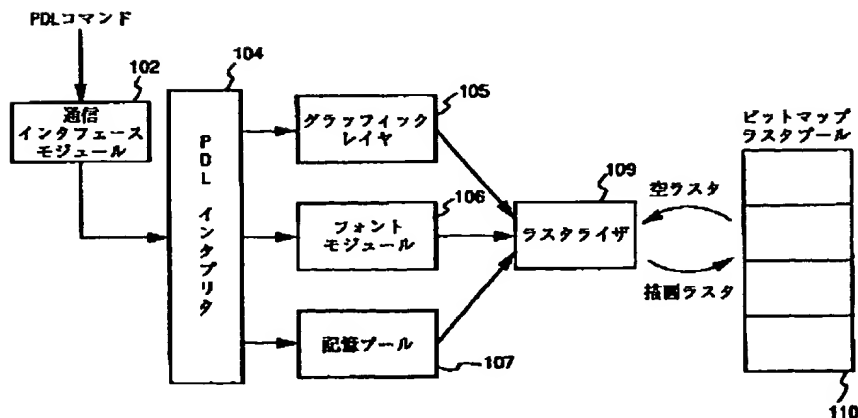
【図1】



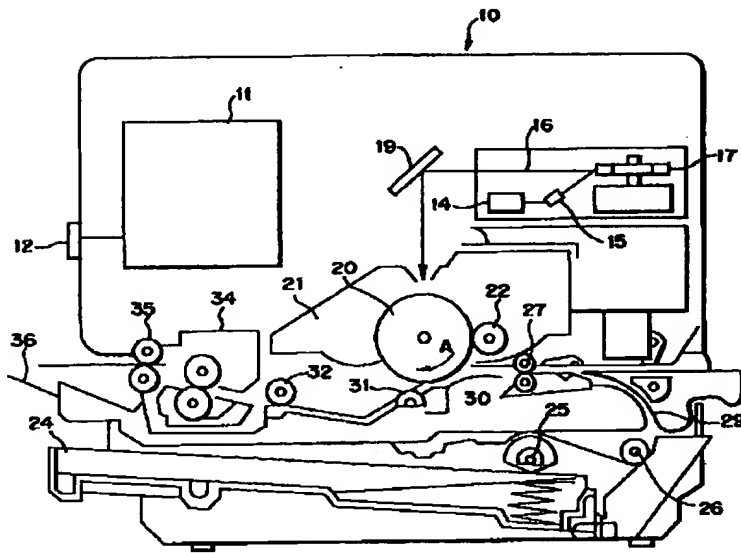
【図6】



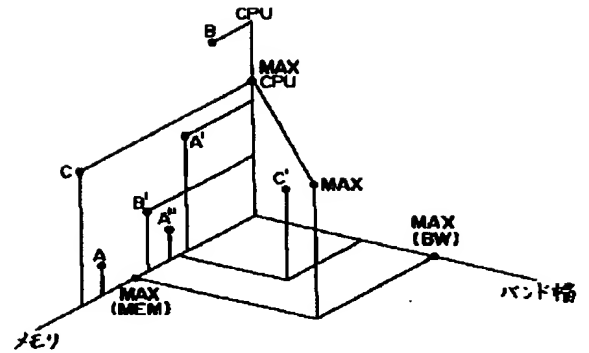
【図2】



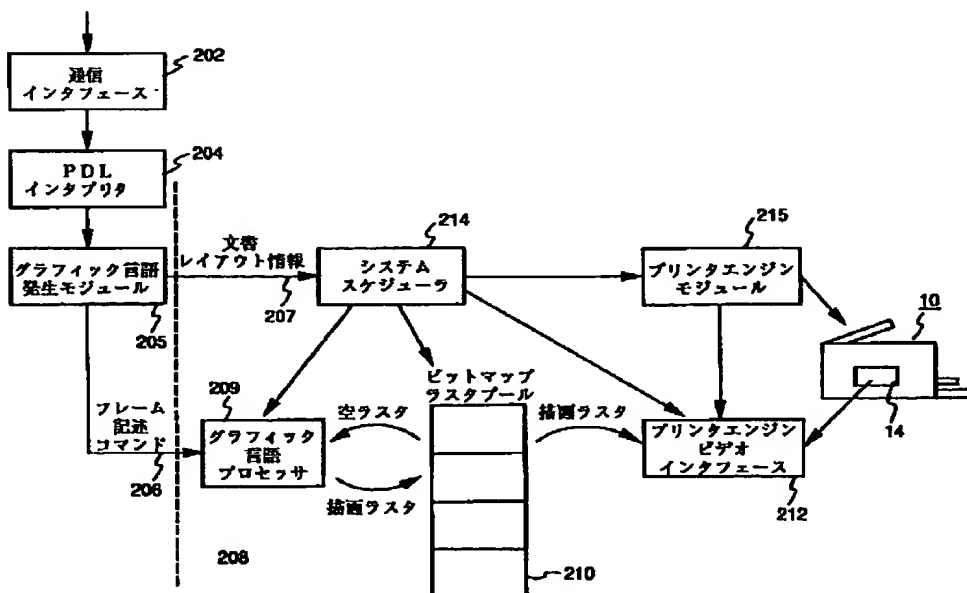
【図3】



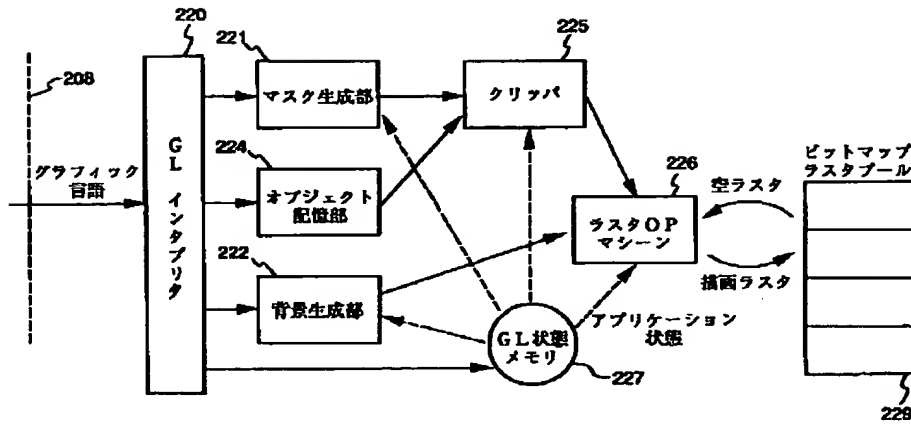
【図12】



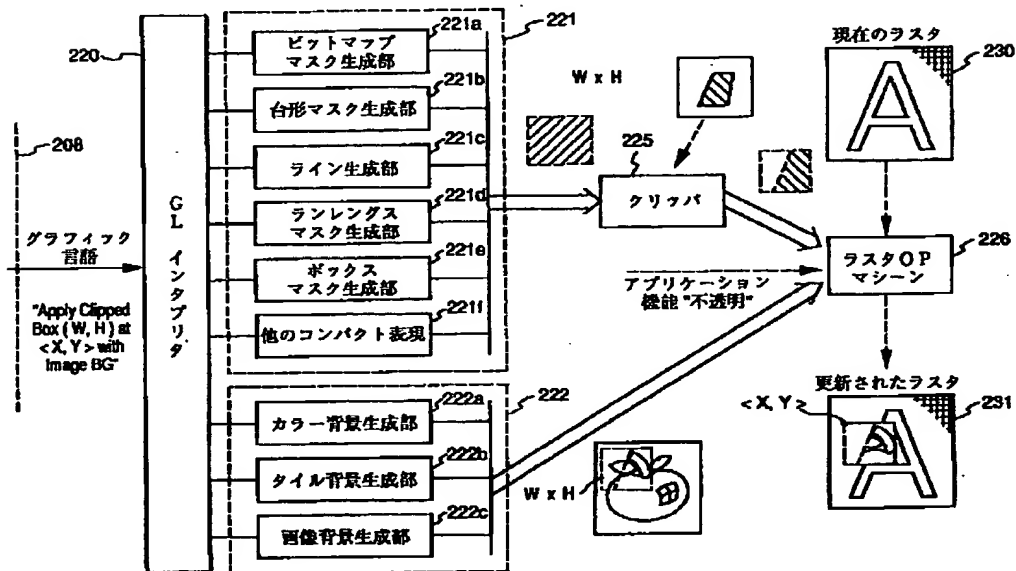
【図4】



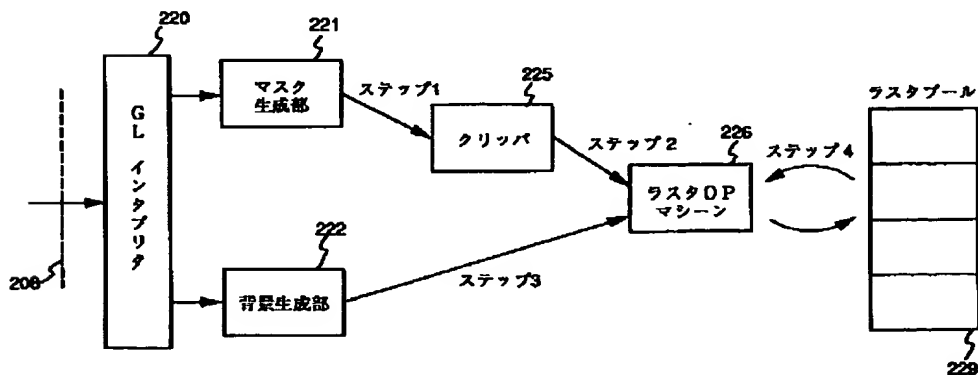
【図5】



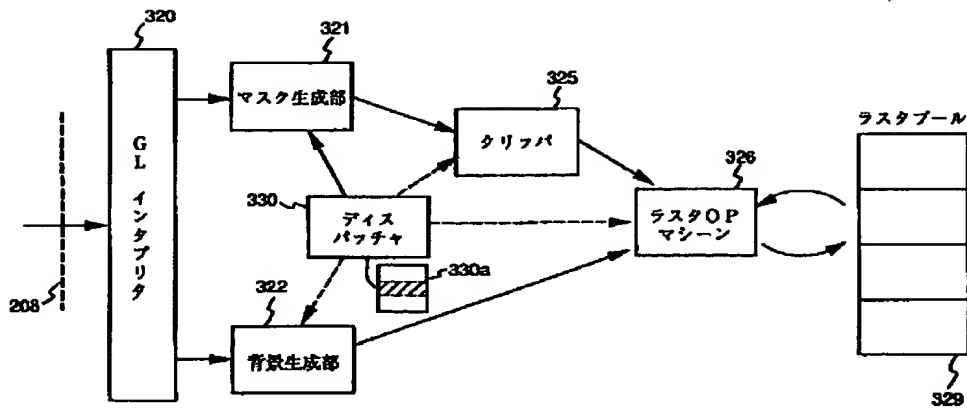
【図7】



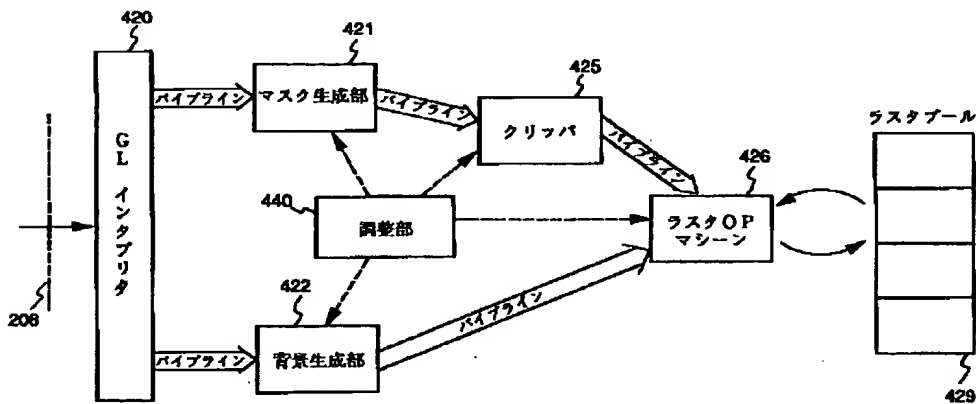
【図8】



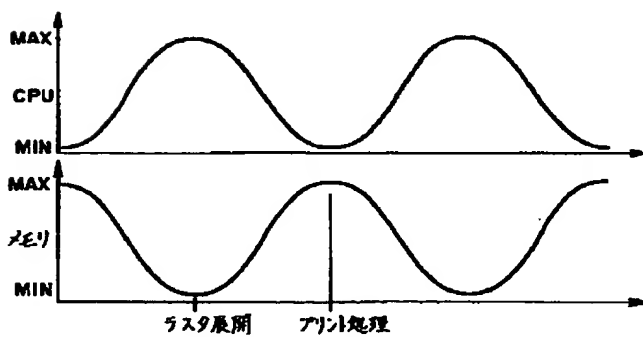
【図9】



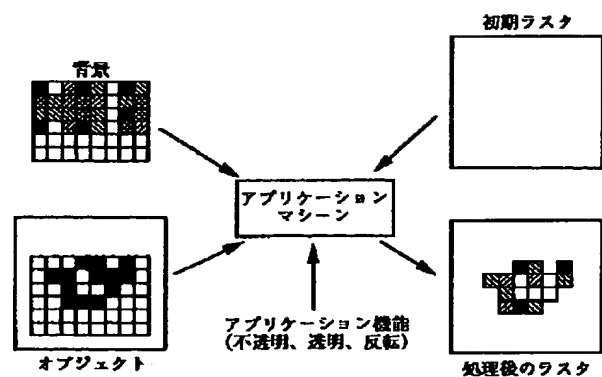
【図10】



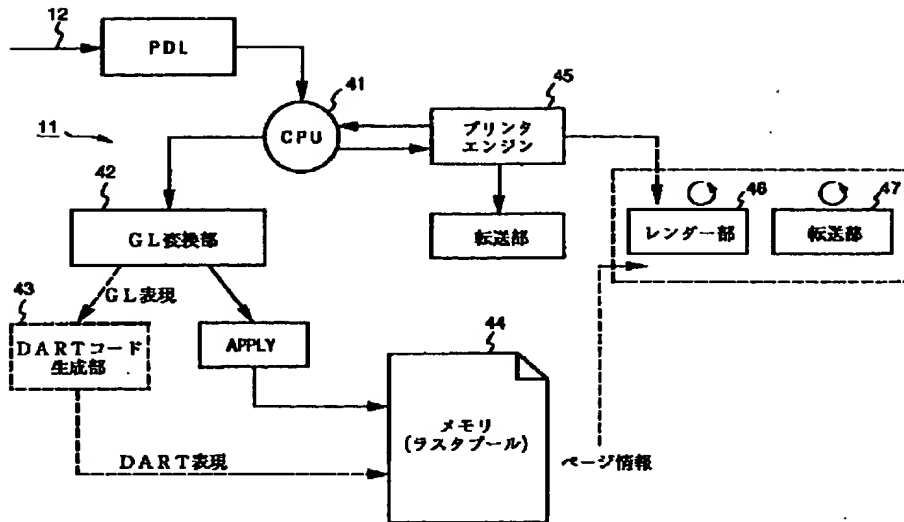
【図11】



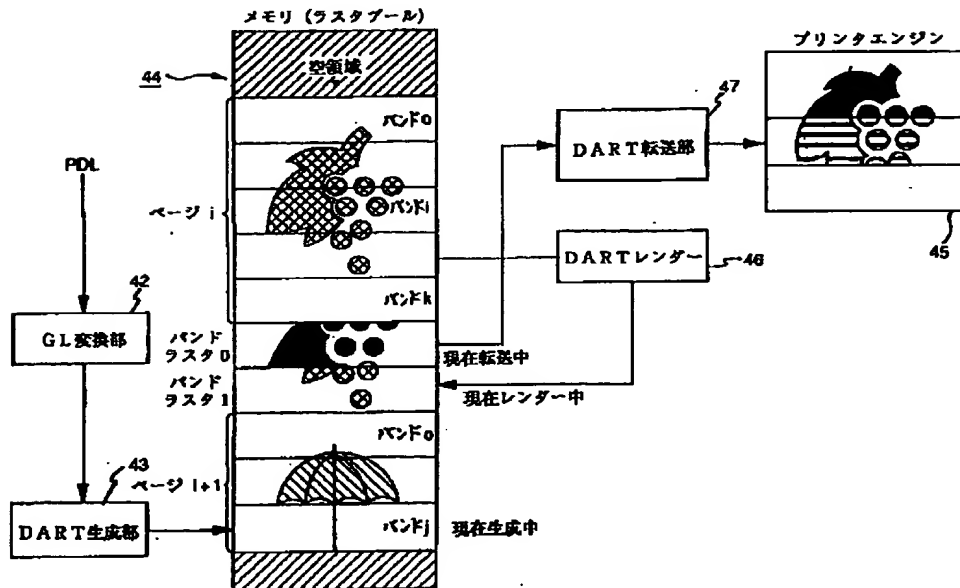
【図17】



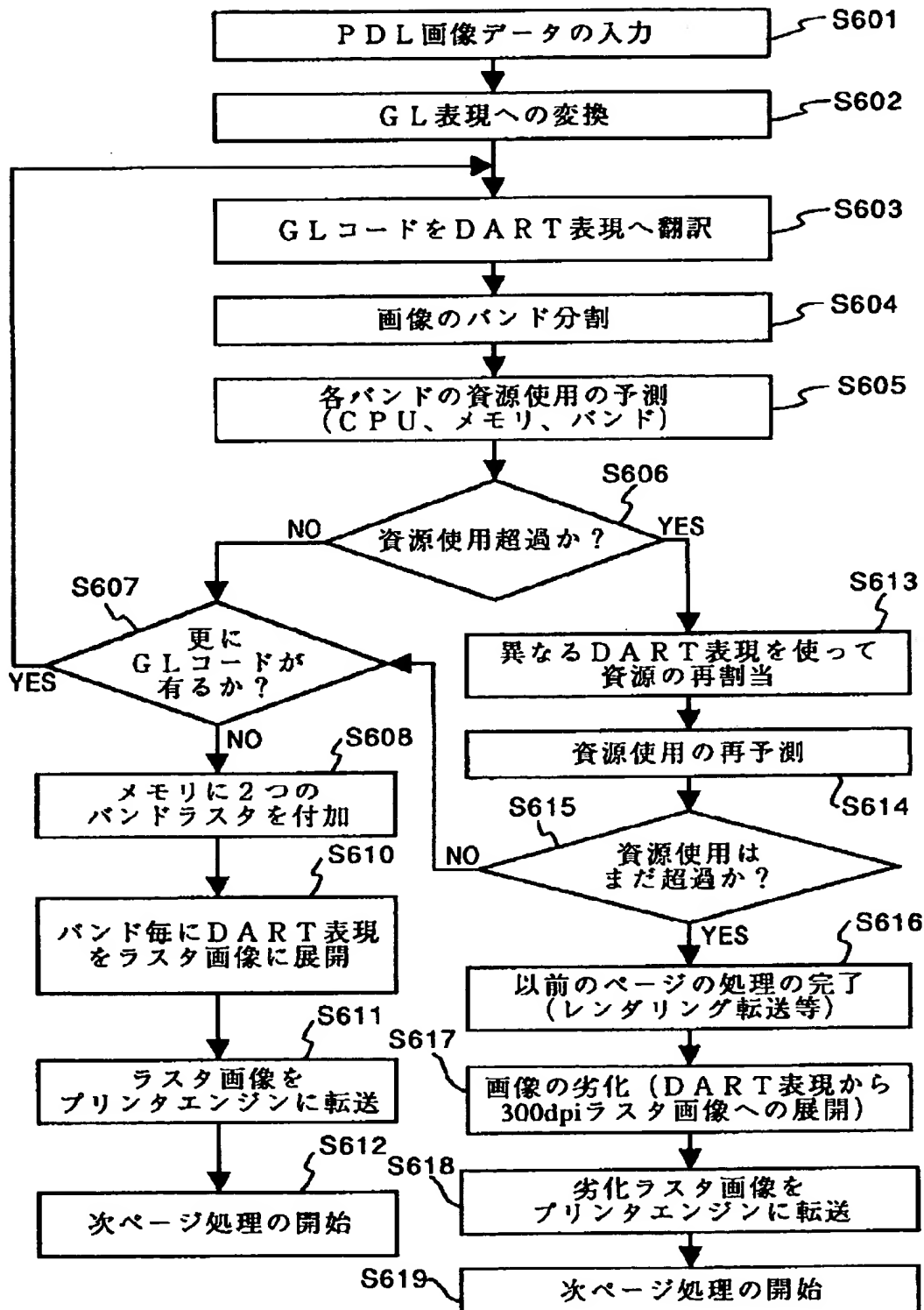
【図13】



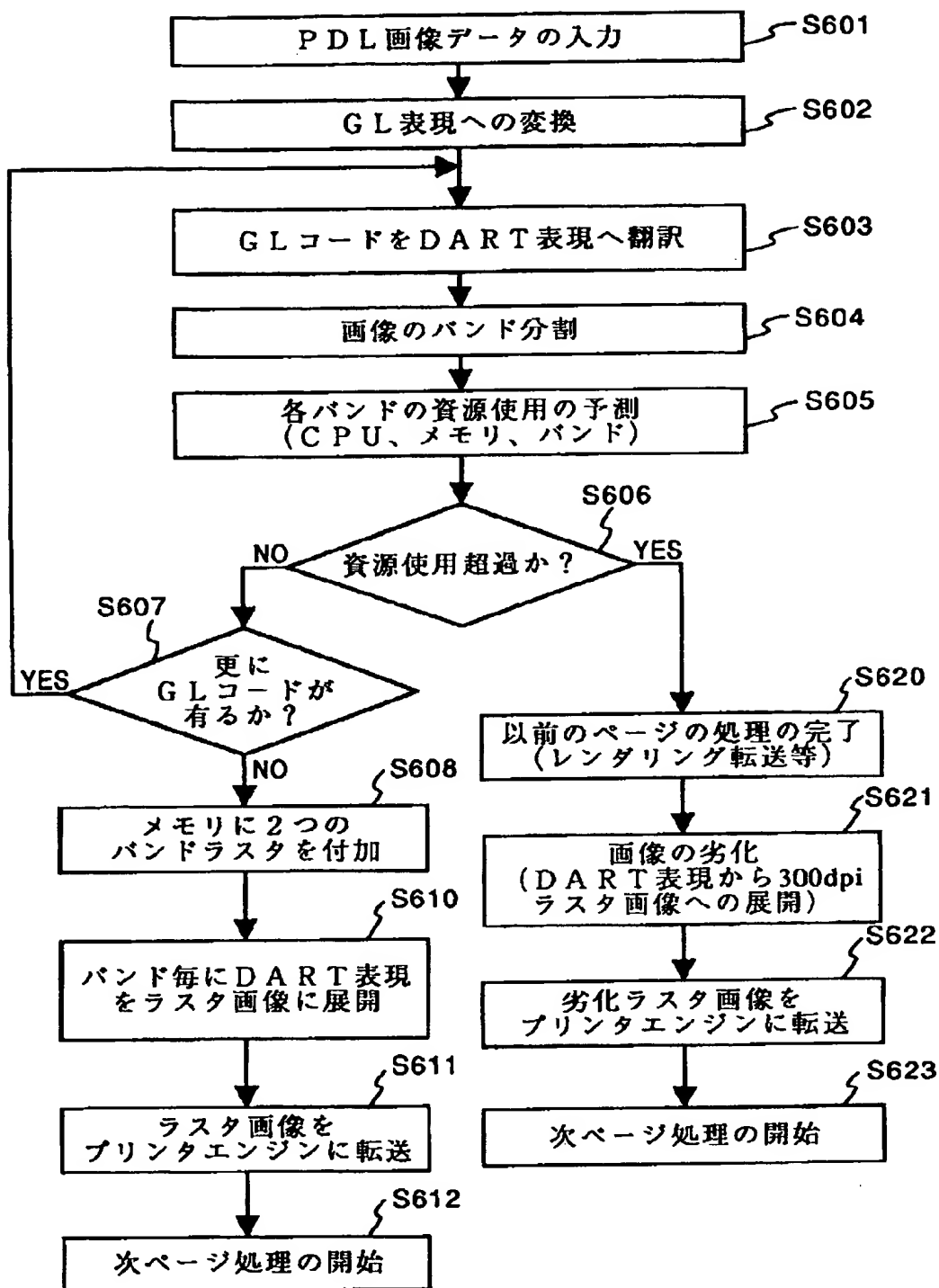
【図14】



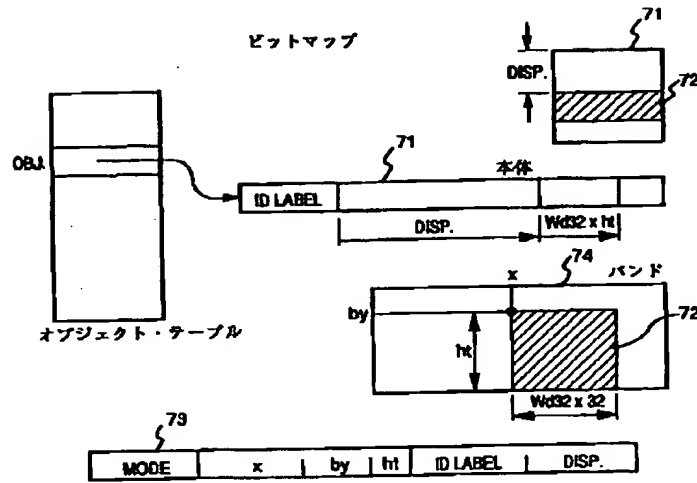
【図15】



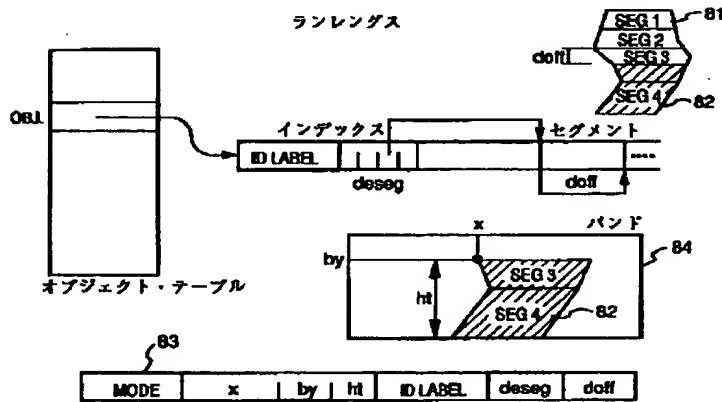
【図16】



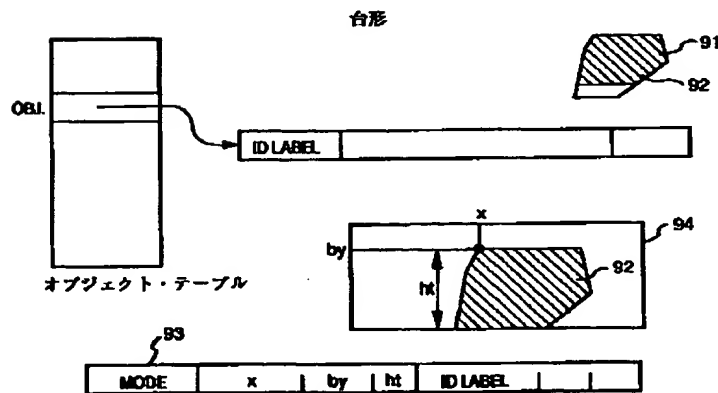
【図18】



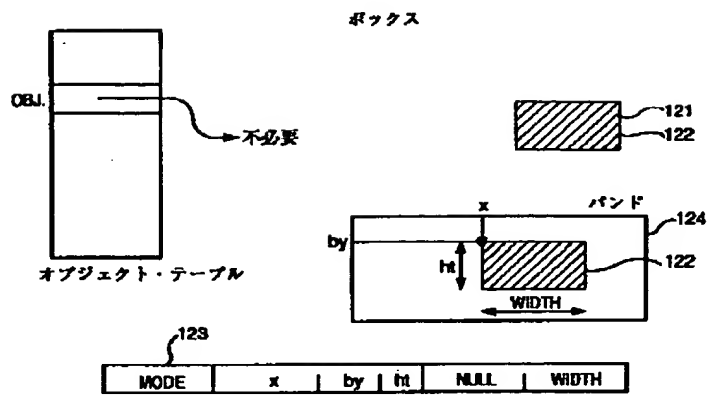
【図19】



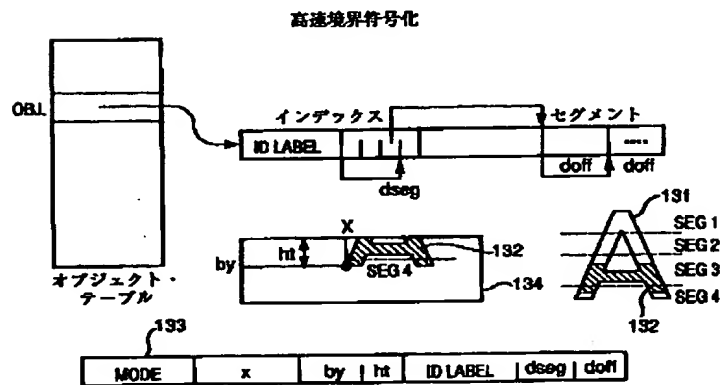
【図20】



【図21】



【図22】



フロントページの続き

(51)Int. Cl.⁶

G 0 6 F 15/72

識別記号 片内整理番号

G 9192-5L

F I

技術表示箇所

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☒ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.